

# Programmazione in PHP e CMS

## Sistemi Distribuiti, Parte 3

Corso di Laurea in Ingegneria o per altri CDL

*Ing. Michela Paolucci*

Department of Information Engineering, DINFO

University of Florence

Via S. Marta 3, 50139, Firenze, Italy

tel: +39-055-2758515, fax: +39-055-2758570

**Lab: DISIT, Sistemi Distribuiti e Tecnologie Internet**

**<http://www.disit.dinfo.unifi.it/>**

[michela.paolucci@unifi.it](mailto:michela.paolucci@unifi.it)

<http://www.disit.org/nesi>, <http://www.dsi.unifi.it/~paolucci>

# Sistemi Distribuiti

**Corso di laurea in Ingegneria**

**Programmare per il Web:  
Il linguaggio PHP**

- 1. Parte I: Introduzione**
- 2. Parte II: PHP**
- 3. Parte III: Approfondimenti**

# Sistemi Distribuiti

## Corso di laurea in Ingegneria

1. **Parte I: Introduzione al Web**
  - perchè nasce il php
  - architettura del web
  - Web server
2. **Parte II: PHP**
3. **Parte III: Approfondimenti**

# PHP: come nasce

- Il World Wide Web è stato creato da Tim Berners-Lee nel 1991
- A metà degli anni Novanta il Web era ancora formato in gran parte da **pagine statiche**
- Con l'evoluzione di Internet, si comincia a sentire l'esigenza di rendere dinamici i contenuti, cioè di far sì che la stessa pagina sia in grado di proporre contenuti diversi, personalizzati in base alle preferenze degli utenti, oppure estratti da una base di dati (database) in continua evoluzione
- Per questo scopo nasce il PHP



# PHP (1)

- PHP nasce nel 1994, ad opera di Rasmus Lerdorf, con lo scopo di facilitare ai programmatori l'amministrazione delle homepage personali
- In seguito venne ampliato fino alla realizzazione di una versione (Form Interpreter, PHP/FI) che prevedeva la possibilità di integrare il codice PHP nel codice HTML in modo da semplificare la realizzazione di pagine dinamiche

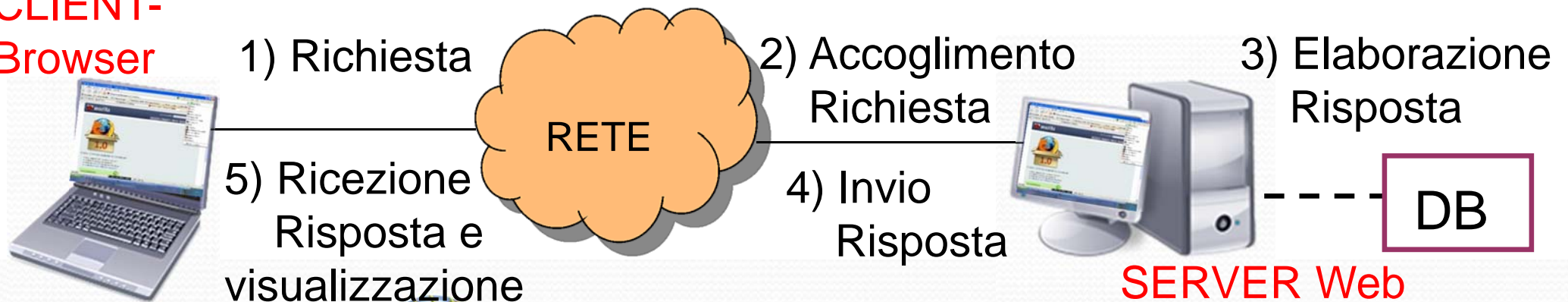
# PHP (2)

- PHP (Hypertext PreProcessor) è un linguaggio di scripting Open Source molto utilizzato, è specialmente indicato per lo sviluppo Web e può essere integrato nell'HTML
- L'obiettivo principale del PHP è quello di permettere agli sviluppatori web di scrivere velocemente pagine web dinamiche, ma con PHP si possono fare molte altre cose
- E' integrabile in numerosi server Web (Apache HTTP Server, Internet Information Services - Microsoft -, .. )

# Architettura Software del Web

- L'architettura software del web può essere divisa in due categorie logiche:
  - Lato client: qualunque dispositivo che si connette alla rete e che comunica attraverso di essa con tutti gli altri dispositivi connessi. Nel client è sufficiente la presenza di un software per la visualizzazione delle informazioni che il server fornisce (Internet Explorer, Netscape, Mozilla, Opera, etc.)
  - Lato Server: è un dispositivo che attende le richieste del client ed invia le relative risposte dopo opportune elaborazioni (es: interrogazione del database).

CLIENT-  
Browser

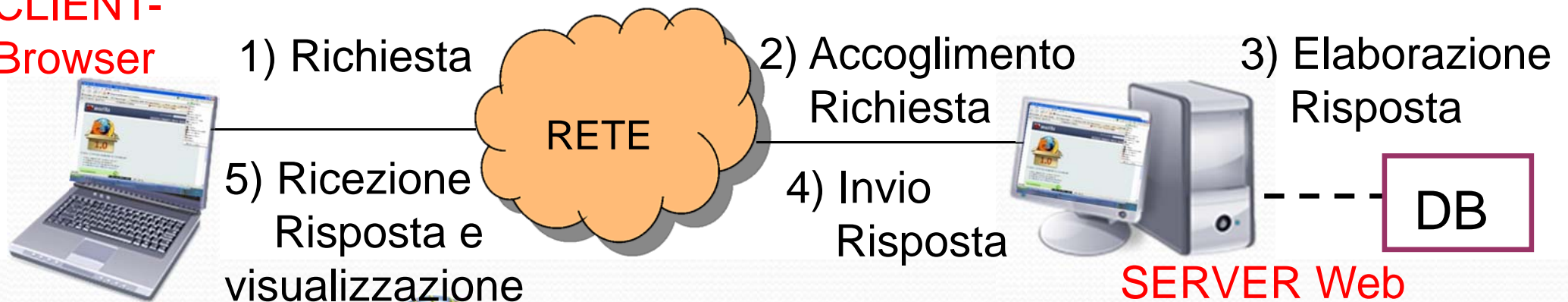




# Lato client: Browser

- Il Browser è un programma usato per navigare nel Web è serve per:
  - scaricare i file che si trovano sui server
  - visualizzare le pagine html
- Oltre ad Internet Explorer, il Browser più diffuso, ne esistono altri: Netscape Navigator, Mozilla, Opera

CLIENT-  
Browser

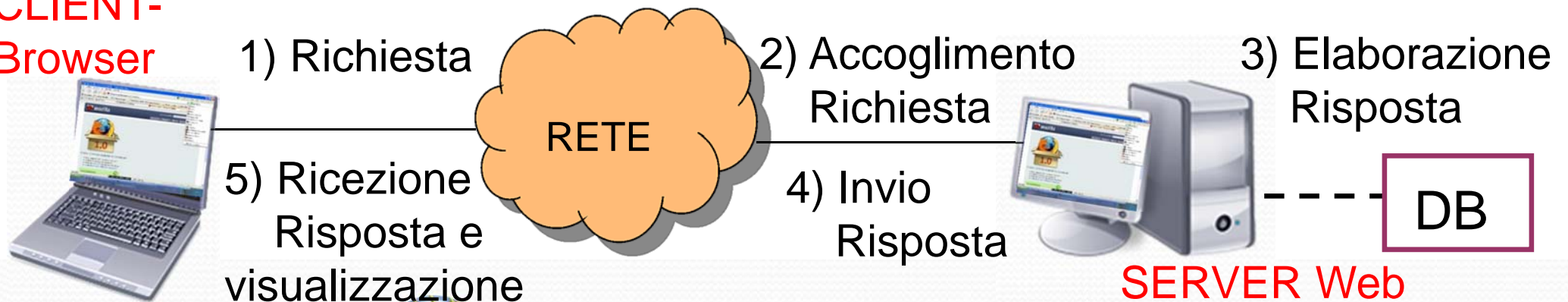




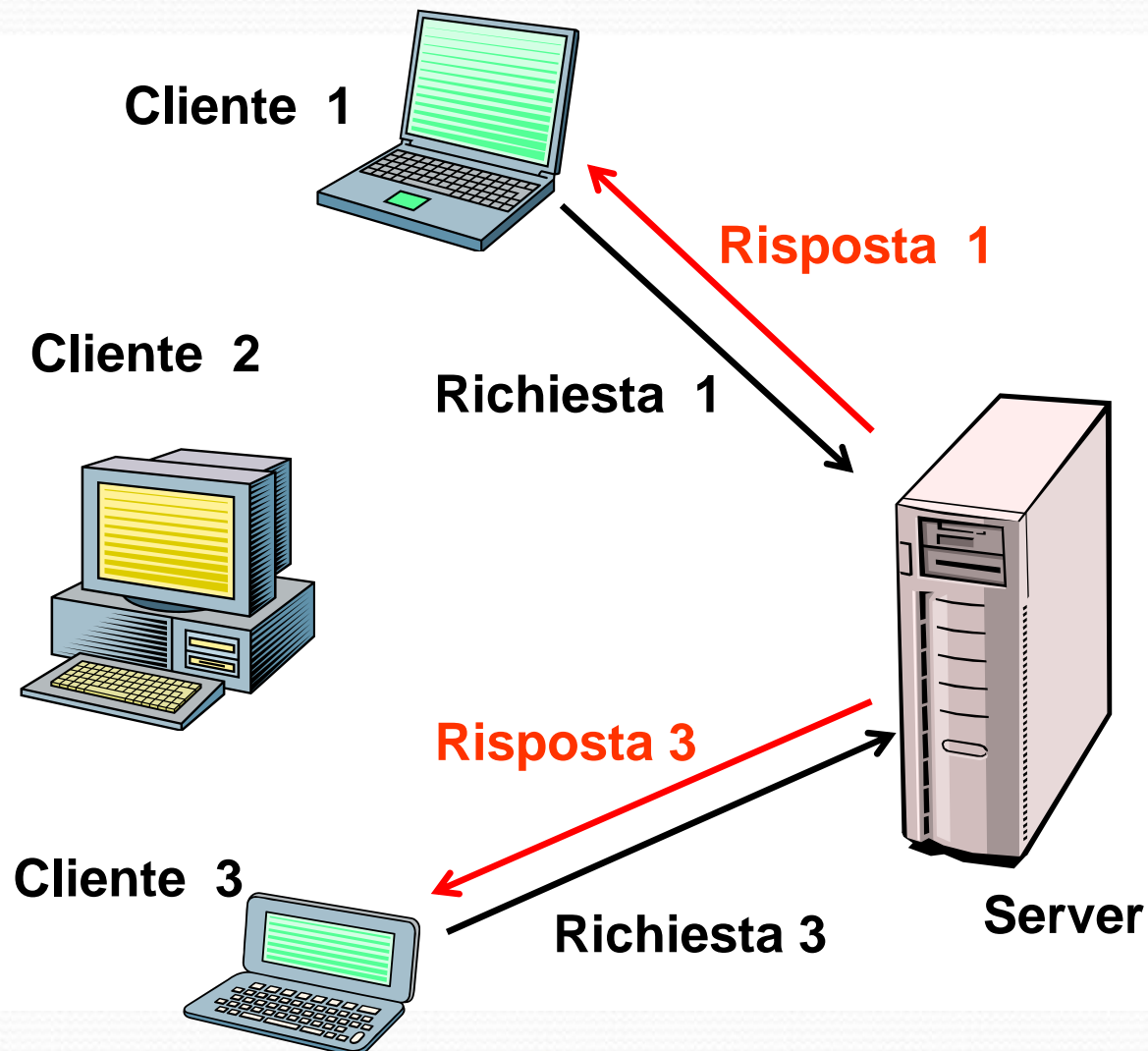
# Lato server: Server Web

- Il Server Web è un software che resta in ascolto delle richieste di accesso ad un sito web (inoltrate da parte di un Browser, ovvero dal client), le processa e restituisce dei dati come risposta
- I dati forniti dal Server contengono gli elementi necessari per la visualizzazione di una pagina web e vengono così analizzati ed interpretati dal Browser che li presenta all'utente nel miglior modo possibile

CLIENT-  
Browser

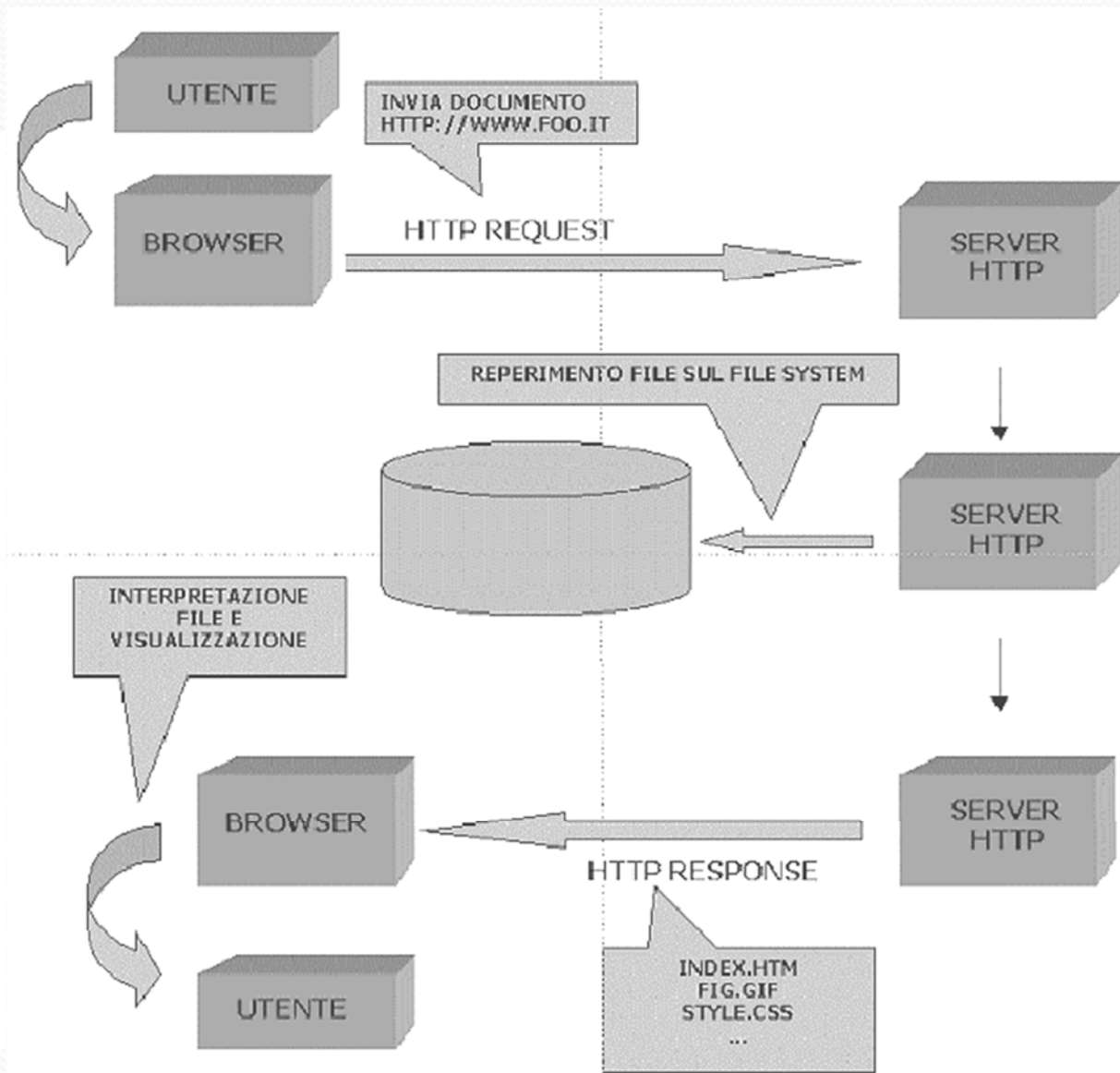


# Architettura client-server



Tutta la comunicazione avviene seguendo regole prestabilite legate al tipo di *protocollo* (ad es. http). Il protocollo fondamentale di Internet, sul quale si basano tutti gli altri, è il **TCP/IP**

# Il Web Server



E' un programma sempre attivo che ascolta su una porta le richieste HTTP. All'arrivo di una richiesta la esegue e restituisce il risultato al browser, rimettendosi in ascolto



# Quali web server?

- Essendo il protocollo HTTP uno standard libero, in linea di principio chiunque è in grado di costruirsi un proprio server web, seguendo il protocollo HTTP.
- I web server sono ormai diventati strumenti estremamente complessi proprio per il supporto alla tecnologia server side, oltre che per ragioni di efficienza nel soddisfare un numero molto alto di richieste contemporanee
- I web server più conosciuti e usati nei siti di tutto il mondo sono:
  - Apache (<http://www.apache.org>)
  - Microsoft Internet Information Server (IIS)



# Gestire un web server

- Scelta del server HTTP
- Principali parametri di configurazione:
  - Avvio, spegnimento
  - Porta di funzionamento
  - Directory di default
  - Documento di default
  - Log degli accessi/errori
  - Alias o cartelle virtuali

# Avvio e spegnimento del server

- Tipicamente uno server web per sua stessa definizione è un programma *sempre attivo*.
- Ci sono però dei casi in cui vogliamo disattivare il server (mettere fuori linea il sito per manutenzione, ad esempio). Per questa ragione ogni server web ha comandi per fermare il server e riavviarlo

# Web Server e porte

- Il web server (o server HTTP) come tutti i servizi basati su TCP/IP, si attiva su una **porta**, che è il numero a cui si fa corrispondere il servizio.
- In questo modo una macchina può far girare più servizi differenziando le porte
- Ci sono porte standard per i vari servizi (ftp, telnet, posta elettronica, web)
- La porta di default del web server è la **80**, ma può essere configurato per funzionare su un'altra porta libera. In questo caso la porta va specificata sia nella configurazione del webserver che nella URL

# Web server e porte

Esempi:

`http://localhost:8080/`

`http://www.miodominio.it:9736/esempi/esercizio.php`

- Negli esempi di queste URL il web server è stato configurato per funzionare sulla porta 8080 (nel primo caso) e sulla 9736 (nel secondo caso).
- Nella prima URL la parola chiave **localhost** sta a indicare il server web attivo sulla macchina locale (può ovviamente funzionare senza una connessione internet), la macchina locale fa da server e da client



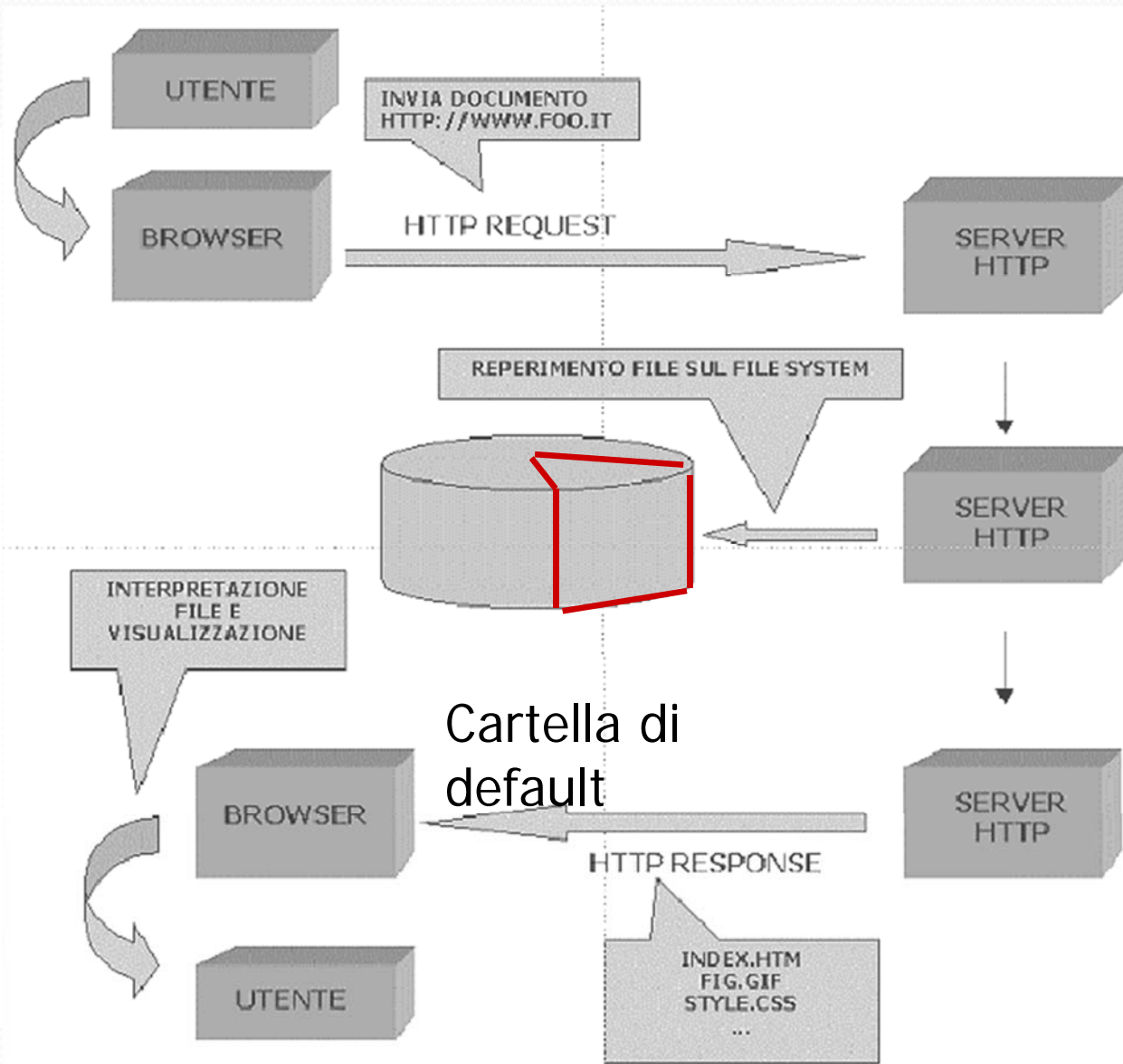
# Web server e documenti

`http://localhost:8080/`

`http://www.miodominio:9736/esempi/esercizio.php`

- Nella prima URL attiviamo una richiesta al server web locale attivo sulla porta 8080. Non specifichiamo quale documento restituirci. Il web server in questo caso decide di ritornarci il **documento di default**
- Nella seconda URL invece specifichiamo di restituirci la pagina **esercizio.php** nella cartella **esempi**

# Web Server e documenti

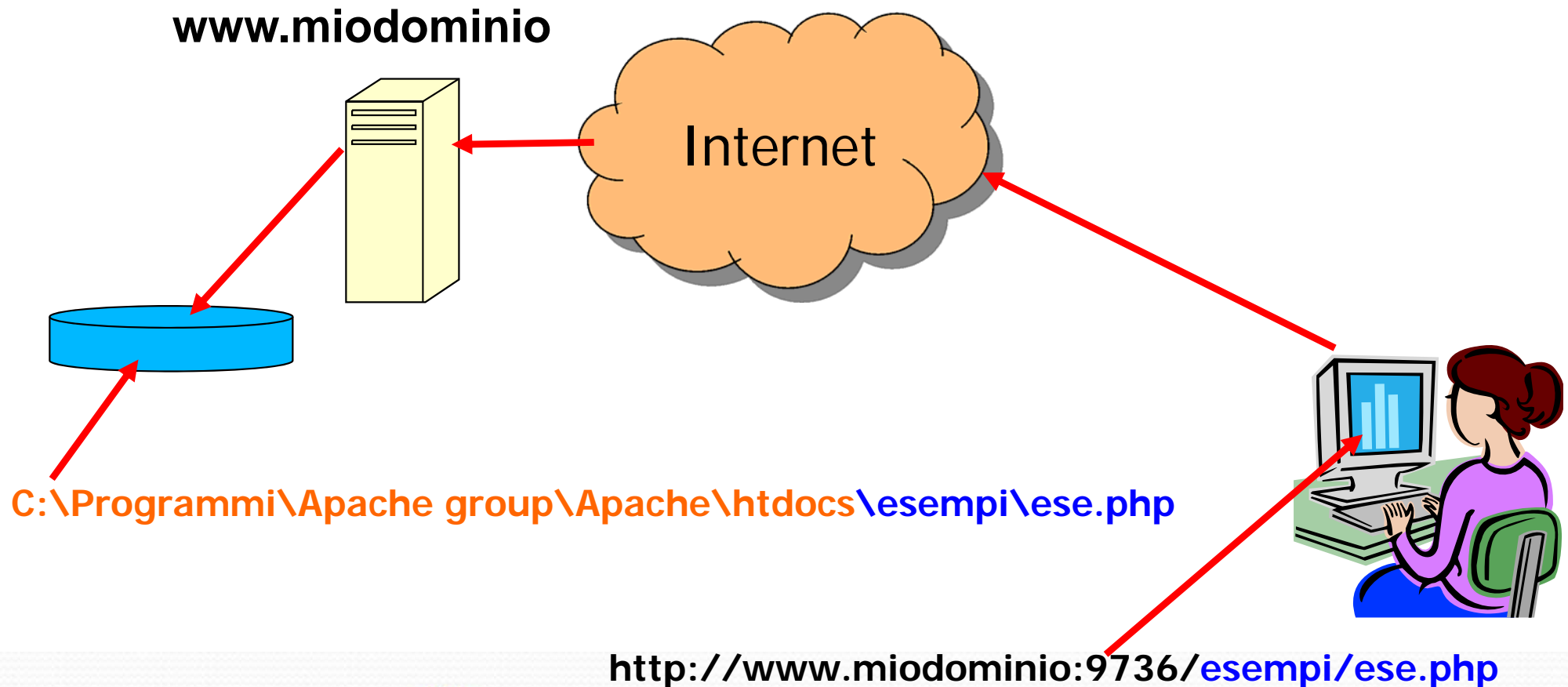


- Il server web ha visibilità di una sola parte del file system della macchina. Per ragioni di sicurezza e di pulizia del sistema
- La porzione di file system visibile dal server web si chiama **directory (o cartella) di default** del server web

# Web server e documenti

- La cartella di default del server web varia a seconda del server considerato e del sistema operativo su cui è installato.
- Ad esempio:
  - Per *Apache* su windows tipicamente è la cartella **htdocs** che si trova nella directory di installazione di Apache (ad es. *C:\Programmi\Apache Group\Apache\htdocs*)
  - Per *IIS* è la cartella **wwwroot** che generalmente si trova in *C:\inetpub\wwwroot*

# URL e directory





# Documenti di default

- E' uso comune configurare il server web affinché restituisca un documento predefinito qualora non ne venga specificato uno nella URL
- Questo permette ai siti web di rendere visibile una “home page” digitando il solo indirizzo del web server
- Ad es: [www.repubblica.it](http://www.repubblica.it) o [www.google.com](http://www.google.com)
- La pagina visualizzata prende il nome di **homepage** e tipicamente ha uno dei nomi: [index.html](#), [index.htm](#), [index.php](#), [default.html](#), [default.htm](#), [default.asp](#) etc...
- Queste pagine di default e l'ordine in cui vengono cercate **possono essere configurate sul server web**

# Apache

- Il web server Apache è free opensource creato e mantenuto dalla Apache foundation ed è liberamente scaricabile da <http://www.apache.org> dove si può trovare tutta la documentazione
- Può essere installato sia su Windows che su Unix anche se è stato realizzato specificatamente per Unix.
- Supporta varie tecnologie *server-side*, occorre però installare il corrispondente modulo. Alcuni moduli sono già installati nella configurazione di default, altri vanno appositamente installati (ad es PHP).
- Esistono installazioni integrate di Apache + PHP + MySQL

# Mettere in linea una app. web

- Nella fase di progettazione della applicazione web una fase importante e' la scelta dell'hosting del sito.
- La scelta può variare tra tenere un proprio server in rete oppure usare lo spazio fornito da un provider (gratuito o a pagamento)
- Queste scelte hanno vantaggi e svantaggi complementari:
  - Il server proprio permette maggiori possibilità di personalizzazioni e flessibilità nella configurazione, ma maggiori costi di mantenimento
  - Il server in hosting scarica il webmaster dal peso dell'installazione e il mantenimento del software e dell'hardware, ma non permette flessibilità e personalizzazione della configurazione
- Molti provider forniscono gratuitamente o a basso costo il supporto per PHP/MySQL



# Client-Server

Browser

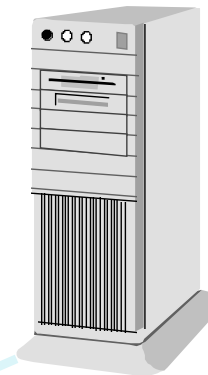


Pagina html



Client

Server web



Server di  
Repubblica  
risponde  
su http

richiesta

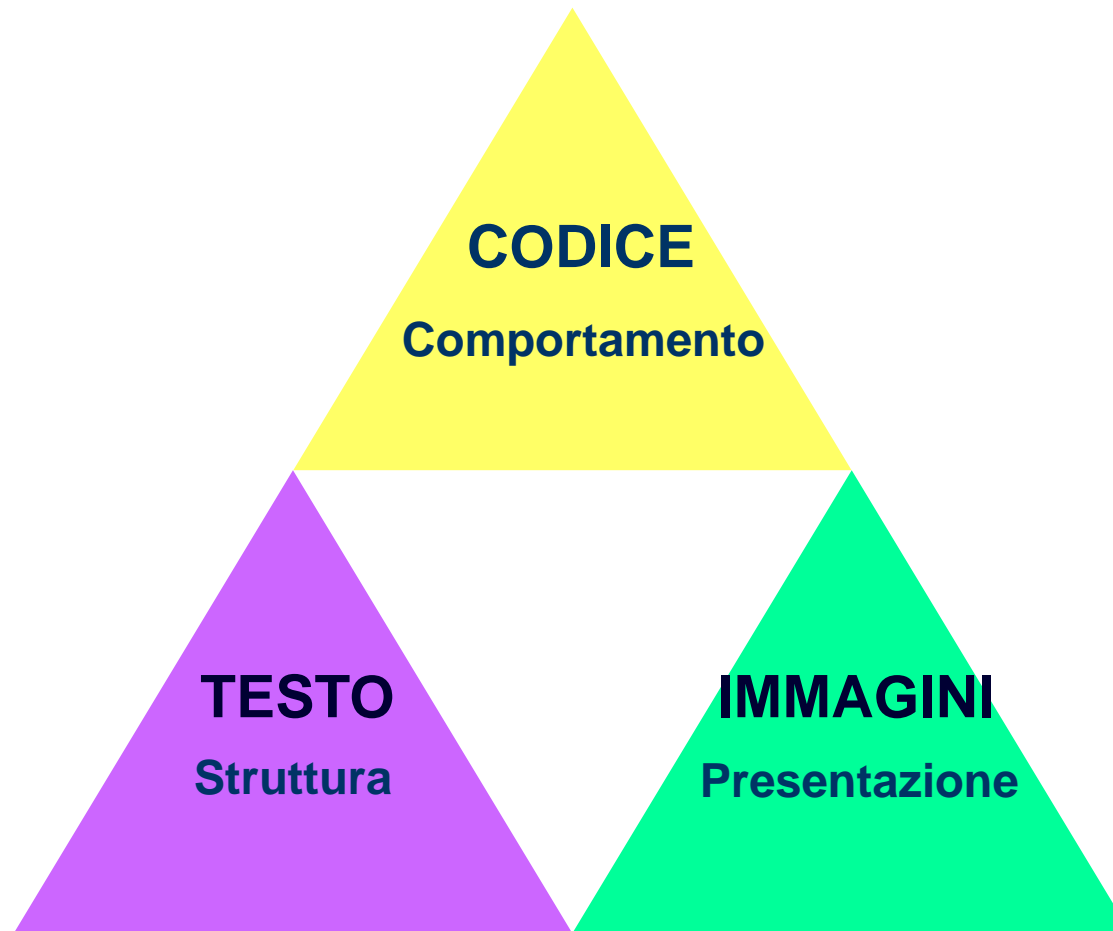
servizio

Protocollo http  
Pagina html

# Un modello concettuale per il Web

- Nell'editoria tradizionale:
  - Testo (contenuti)
  - Immagini (presentazione visuale)
- Il Web aggiunge il comportamento
  - Si seguono collegamenti ipertestuali
  - Si fanno ricerche
  - Si immettono informazioni
  - Si consultano cataloghi
  - Si effettuano pagamenti
  - ...

# Un modello concettuale per il Web





# Sistemi Distribuiti

## Corso di laurea in Ingegneria

**Programmare per il Web:**

**Il linguaggio PHP**

- 1. Parte I: Introduzione**
- 2. Parte II: PHP**
- 3. Parte III: Approfondimenti**

# PHP: funzionamento generale

- PHP può essere utilizzato inserendo codice PHP in mezzo al codice HTML che compone una pagina web
- Quando un utente (client) di un sito web (gestito dal server) arriva ad una pagina che contiene codice PHP, il server stesso lo esegue. In pratica il PHP viene eseguito localmente sul server che gestisce il sito web
- Non è possibile eseguire il codice PHP su un computer che non sia anche un server
- PHP è un linguaggio di scripting, ovvero non viene compilato prima di funzionare ma in tempo reale (mentre l'utente fa le richieste al server)
- Concludendo: I file *.php* possono contenere parti scritte in HTML e parti scritte in PHP delimitate da tag particolari. L'interprete php si occupa di gestire tutto il codice compreso all'interno di tali tag

# Tag di inizio e fine del codice php

- Esistono 4 set di tag che possono essere usati per delimitare blocchi di codice php. I primi due sono quelli standard:
  - **<?php echo 'Testo'; ?>**
  - **<script language='php'> echo (Testo); </script>**
  - **<? echo 'Testo'; >**
  - **<% echo (Testo); %>**

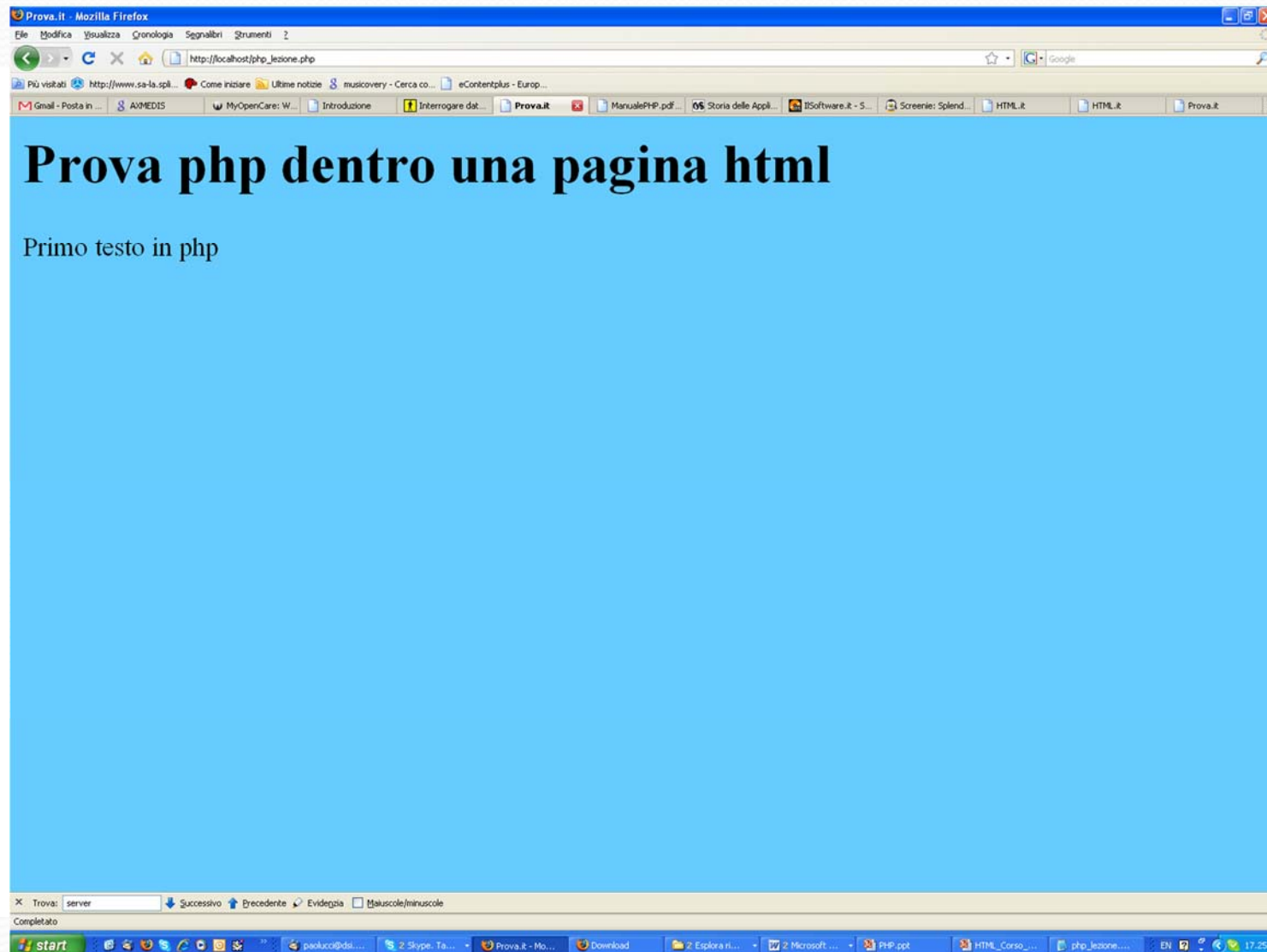


# Esempio: codice php in una pagina html

NOTA: nomefile.php contiene il seguente codice:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-1">
    <title>Prova.it</title>
  </head>
  <body bgcolor="#66CCFF">
    <h1>
      Prova php dentro una pagina html
    </h1>
    <?php
      echo '<b>Primo testo</b> in <i>php</i>';
    ?>
  </body>
</html>
```

# Visualizzazione Esempio: codice php in una pagina html



# I Commenti

- Esistono più metodi:
  - Commento fino a fine riga:  
**# commento 1**  
**// commento 2**
  - Commento su più righe:  
**/\* Questo è il**  
**commento 3**  
**\*/**



# Variabili

- Le variabili PHP possono contenere array, stringhe, numeri, etc.
- Le variabili **non vanno dichiarate** e non hanno un tipo prefissato (possono cambiare tipo durante l'esecuzione)
- L'assegnazione viene fatta tramite l'operatore '='
- È possibile assegnare dei valori alle variabili:
  - Per valore: il valore della variabile di origine viene copiato in quella di destinazione

**\$nomeVariabile = valore;**

- Per riferimento: invece di creare una copia della variabile, si effettua un puntamento alla locazione di memoria della variabile. Ne consegue che modifiche sul valore di una variabile si ripercuotono sull'altra.

**\$nomeVariabile = &nomeAltraVariabile;**

- ```
<html>
  <body>
    <?php
      $myVar = "La mia prima variabile";
      echo $myVar;
    ?>
  </body>
</html>
```

# Nomi delle variabili: sintassi

- I nomi delle variabili sono 'case sensitive'
- I nomi delle variabili si possono scegliere usando lettere, numeri e 'underscore' (\_)
- Il primo carattere usato NON può essere un numero (si usano lettere o underscore)
- I nomi delle variabili sono precedute dal simbolo '\$'
- Es:
  - \$a
  - \$nome
  - \$p1
  - \$nome\_cognome

# Tipi di Dati

- Php supporta I seguenti tipi di dati:
  - boolean
  - integer
  - float
  - string
  - array
  - object
  - resource
  - NULL
  - mixed
  - number
  - callback
  - ...

# Tipi di Dati: boolean

- Nota: tutte le variabili in PHP devono iniziare con il simbolo \$
- Il **boolean** è il tipo di dato più semplice, può assumere due valori: True o False

**<?php**

**\$variabileVero = True;**

**\$variabileFalso = False;**

**?>**



# Tipi di Dati: integer

- Un **Integer** è un numero intero positivo o negativo di lunghezza dipendente dal sistema operativo, in genere 32 bit con segno, può essere specificato in base 10, 16, 8 eventualmente preceduto dal segno. I numeri che iniziano con una cifra diversa da 0 vengono considerati decimali, se iniziano con 0 ottali, se iniziano con 0x esadecimali.
- **<?php**
  - \$a = 1234; # numero decimale**
  - \$a = -123; # numero negativo**
  - \$a = 0123; # numero ottale (83 decimale)**
  - \$a = 0x1A; # numero esadecimale (26 decimale)**

**?>**

# Tipi di Dati: float

- **float**, anche in questo caso la dimensione del dato dipende dal sistema operativo, in genere 64 bit. Assume valori decimali a virgola mobile:

```
<?php
```

```
    $myDouble = 10.341;
```

```
    $myDouble = 0.14e2;
```

```
?>
```

# Tipi di Dati: string

- **string** contiene testo. Si hanno due modalità:

- Tra apici ('testo'), in questo caso se si vuole inserire un apice nella stringa, è necessario farlo precedere da backslash (\) , il carattere backslash può essere inserito raddoppiandolo(\\)

```
<?php echo 'testo in formato \'txt\' ' ; ?>
```

(visualizza: testo in formato 'txt')

- Tra virgolette ("testo"), in questo caso si possono usare i caratteri speciali del linguaggio di programmazione C (\n,\r,\\,\t, ... ) e si può includere il contenuto di altre stringhe:

- "<tr><td>\$nome</td><td>\$cognome</td></tr>"
- "\${importo}euro"

# Tipi di Dati: array

- Un array, contiene una serie di valori accessibili tramite un indice
- Sintassi:

```
array( [key => ] value,...);
```

dove **key** può essere un intero o una stringa e **value** un qualsiasi valore, compreso un altro array.

```
<?php
```

```
$arr1 = array("nome" => "mario", "eta" => 26); //con chiave
```

```
$arr2 = array ('uno', 'due', 'tre', 'quattro', 'stop', 'cinque'); //senza chiave
```

```
echo $arr1["nome"]; // mario
```

```
echo $arr1["eta"]; // 26
```

```
echo $arr2[1]; // due
```

```
?>
```

- Se non specificate alcuna chiave, viene preso il massimo indice intero aumentato di 1 (0 se non ci sono indici interi), se specificate una chiave che esiste già il valore è sovrascritto.

```
<?php
```

```
// Questi due array sono uguali
```

```
$a1=array(5 => 43, 32, 56, "b" => 12, 69);
```

```
$a2=array(5 => 43, 6 => 32, 7 => 56, "b" => 12, 8 => 69);
```

```
?>
```

- La funzione **count(\$arr)** restituisce il numero di elementi nell'array



# Esempio: array

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Prova.it</title>
  </head>
  <body bgcolor="#66CCFF">
    <h1>
      Prova php dentro una pagina html
    </h1>
    <?php
      # come trattare l'apice
      echo '<p>testo in formato \'txt\'<p>';
      // dichiarazione dell'array 'arr'
      $arr = array("id1" => "testo", 12 => true);
      /*
        stampa sulla pagina dei valori dell'array
      */
      echo 'Valore del primo elemento dell\'array: ' ;
      echo $arr["id1"]; // " testo "
      echo '<p>Valore del secondo elemento dell\'array: ' ;
      echo $arr[12];    // 1
      echo '</p>';
    ?>    </body> </html>
```

# Visualizzazione array



# Tipi di Dati: object

- **object**, Il PHP consente di creare oggetti e di usarli in modo simile a C++ e Java, per esempio:

```
<?php
```

```
class oggetto { //definizione della classe
```

```
function metodo(){
```

```
    echo "Ho eseguito una istanza della classe oggetto";
```

```
    } //fine funzione
```

```
} //fine classe
```

```
$istanzaoggetto = new oggetto; //creo il nuovo oggetto
```

```
$istanzaoggetto->metodo(); // e eseguo il suo metodo
```

```
?>
```

# Visualizzazione: object





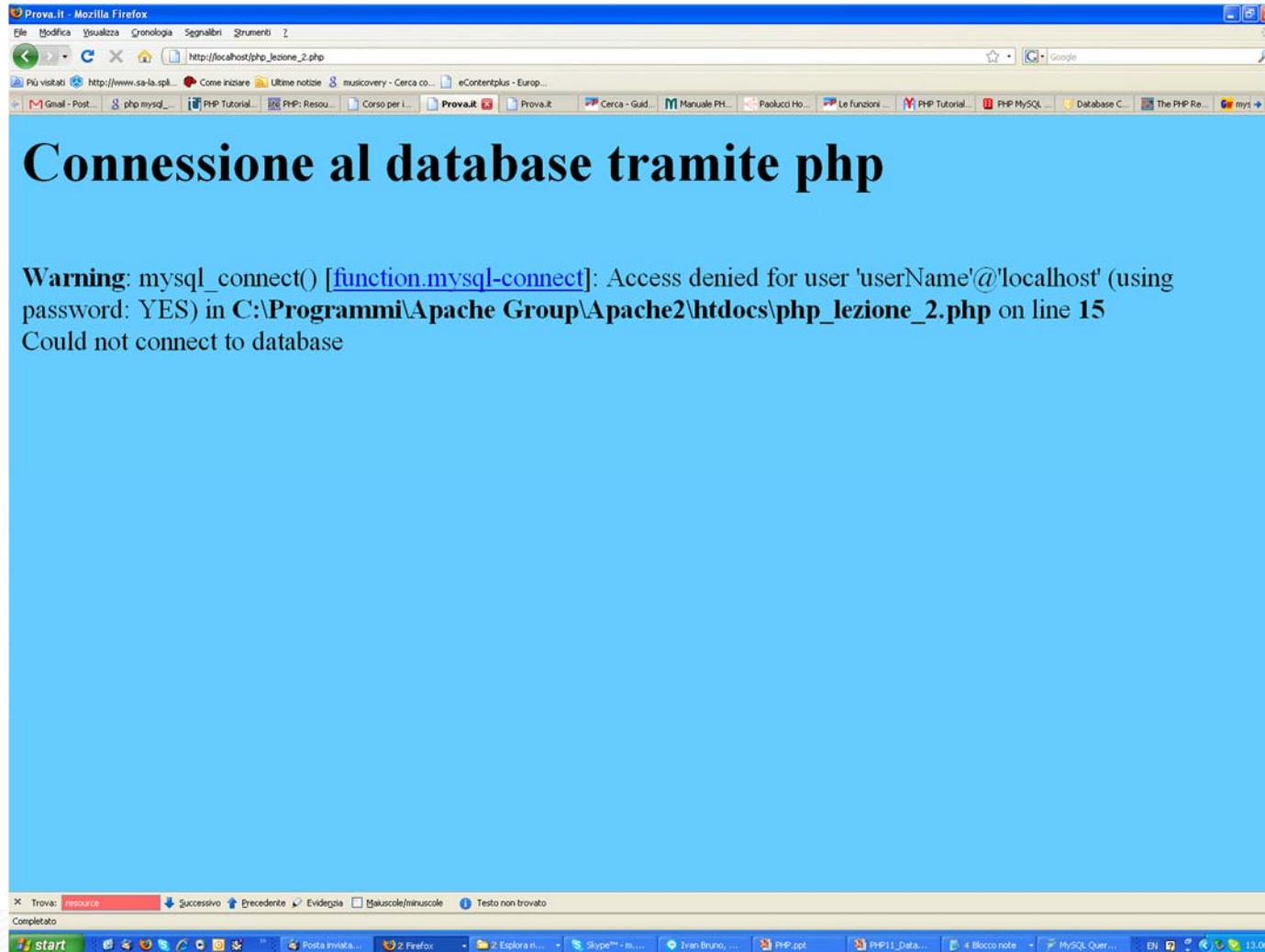
# Tipi di Dati: resource

- **resource**, si usa quando si fa riferimento ad una risorsa esterna (un file, una connessione ad un database, etc.)
- Esempio: resource **mysql\_connect**

```
<body bgcolor="#66CCFF">
<h1>
Connessione al database tramite php
</h1>
<?php
$con = mysql_connect("localhost","UserName","Password");
if (!$con) { echo 'Could not connect to database'; }
else {
    echo '<br>','La connessione al database è andata a buon fine.','<br>',
        'Quello che segue è il contenuto di un campo dell'elemento del database
        selezionato dalla QUERY:', '<br>', 'SELECT * FROM axdbv4.dcmi where id=3';

    $query = "SELECT * FROM dcmi where id=3;";
    $db = 'axdbv4';
    $mysql_result = mysql_db_query ($db, $query, $con);
    // Display the data returned by the query
    while ($row = mysql_fetch_row($mysql_result)) {
        echo '<br>','<br>','Campo: ', $row[1], '<br>';
    }
    mysql_close($con);
}
?> </body>
```

# Visualizzazione: resource errore nella connessione al DB



# Visualizzazione: resource connessione andata a buon fine





# Tipi di Dati: NULL

- Il valore NULL indica che ad una variabile non é stato assegnato nessun valore
- Una variabile é nulla se:
  - Gli è stata assegnata la costante NULL (`$var = NULL;`)
  - Non è stata mai 'settata' (non gli è stato assegnato nessun valore)
  - E` stata usata la funzione `unset()` (`unset(var);`)



# Conversioni tra tipi (esempi)

- In php è possibile trasformare una variabile da un tipo ad un altro:
  - Se abbiamo:
    - **`$a = 5; //intero`**
    - **`$b = '5'; //stringa`**
    - **`$c = 'f'; //stringa`**
    - **`$d = true; //boolean`**
  - Allora:
    - **`$e = $a.$b; //è uguale alla stringa '55'`**
    - **`$f = $a + $b; // è uguale al numero intero 10`**
    - **`$g = $a + $c; // è uguale al numero intero 5`**
    - **`$h = $a.$d; // è uguale alla stringa '51'`**
    - **`$i = $a+ $d; // è uguale al numero intero 6`**

# Conversioni tra tipi

- Si deduce che le conversioni:
  - Da intero a stringa: si ottiene il carattere corrispondente al numero
  - Da stringa a intero: si ottiene, se esiste, un numero corrispondente alla stringa altrimenti 0
  - Da intero a boolean: si ottiene false se è 0 (o minore di 0) e true se è maggiore di 0
  - Da boolean a intero: si ottiene 0 se è false e 1 se è true
  - Da stringa a boolean: si ottiene false se la stringa è vuota e true negli altri casi
  - Da boolean a stringa: si ottiene una stringa vuota se è false, il carattere '1' se è true

# Costanti

- Il PHP mette a disposizione delle costanti, ossia dei valori che si possono usare, ma non modificare (una costante rimarrà invariata per sempre)
- Le costanti non contengono il simbolo '\$' e sono "case-sensitive" per default

```
define("GENNAIO",1);  
define("FEBBRAIO", 2);  
...  
if(m==GENNAIO || m==FEBBRAIO)  
{  
...  
}
```

# Operatori

- Assegnazione
- Operatori aritmetici
- Incrementi
- Operatori per stringhe
- Operatori di confronto
- Operatori Logici
- ...



# Operatori di Assegnazione

- Sono usati per assegnare il valore di una variabile

Operatore	Descrizione/Esempio
=	$x=y$
+=	$x+=y$ equivale a $x=x+y$
-=	$x-=y$ equivale a $x=x-y$
*=	$x*=y$ equivale a $x=x*y$
/=	$x/=y$ equivale a $x=x/y$
%=	$x\%=y$ equivale a $x=x\%y$

# Operatori Aritmetici

Operatore	Descrizione	Esempio
+	Somma	$\$a + \$b$
-	Sottrazione	$\$a - \$b$
*	Moltiplicazione	$\$a * \$b$
/	Divisione	$\$a / \$b$
%	Resto della divisione intera	$\$a \% \$b$
++	Incremento	$\$a++$
--	decremento	$\$a--$

# Operatori per stringhe

- Ci sono due operatori per stringhe:
  - Il primo è l'operatore di concatenazione ('.'), che restituisce la concatenazione dei suoi argomenti a destra e a sinistra:  
**\$a = "Ciao ";**  
**\$b = \$a . "Mondo!"; // ora \$b contiene "Ciao Mondo!"**
  - Il secondo è l'operatore di assegnazione concatenata ('.='), che aggiunge alla fine dell'argomento sul lato destro l'argomento sul lato sinistro:  
**\$a = "Ciao ";**  
**\$a .= "Mondo!"; // ora \$a contiene "Ciao Mondo!"**

# Operatori di confronto

Operatore	Descrizione	Esempio
==	Uguale	\$a == \$b
!=	Diverso	\$a != \$b
>	Maggiore di	\$a > \$b
<	Minore di	\$a < \$b
>=	Maggiore o uguale	\$a >= \$b
<=	Minore o uguale	\$a <= \$b
?:	Operatore ternario	\$ris = (esp1) ? (esp2) : (esp3); \$ris vale esp2 se esp1 è true, altrimenti vale esp3
===	Uguali e dello stesso tipo	\$a === \$b
!==	Non uguali o di tipo diverso	\$a !== \$b



# Operatori Logici

Operatore	Descrizione	Esempi
&& , and	and	$a > 0 \ \&\& \ b > 0$ $a > 0 \text{ and } a < 20$
, or	or	$a \    \ b$ $a < 20 \text{ or } a > 50$
!	not	$! \ a > 70$

# Operatori & Tipi

- Per operatori numerici e di confronto, operandi di tipo diverso sono trasformati di tipo in modo da poter effettuare l'operazione:

Tipo	Tipo	Conversione
intero	float	intero $\rightarrow$ float
intero	string	string $\rightarrow$ int/float
float	string	string $\rightarrow$ float

- Es:
  - $20 > "10"$  (vero, "10" trasformato in 10)
  - $20 + "10"$  (30, "10" trasformato in 10)
  - $150 == "150"$  (vero, "150" trasformato in 150)

# Istruzioni

- Uno script PHP è costituito da una serie di istruzioni
- Una istruzione può essere un'assegnazione, una chiamata di funzione, un ciclo, ...
- Le istruzioni terminano con un punto e virgola
- Le istruzioni si possono raggruppare in blocchi di istruzioni racchiudendole tra parentesi graffe
- Un gruppo di istruzioni è, a sua volta, un'istruzione

# Espressioni

- In php una espressione è una qualsiasi combinazione di funzioni (che vedremo), valori e operatori, che si risolvono in un valore
- Esempi:
  - `15 * 3;` //espressione il cui valore è 45
  - `'Giacomo' . ' Verdi';`  
//espressione il cui valore è 'Giacomo Verdi'
  - `$a + $b;` /\*espressione il cui valore è dato dalla somma dei valori delle variabili \$a e \$b\*/



# Funzioni

- Una funzione è un blocco di codice che può richiedere uno o più parametri in ingresso e può fornire un valore di uscita
- PHP mette a disposizione numerose funzioni predefinite, di cui non si ha sottomano il codice ma risultano molto utili o talvolta indispensabili nella programmazione delle nostre applicazioni server
- In PHP la maggior parte delle funzioni restituisce un valore anche quando ciò potrebbe non essere ovvio: spesso, ad esempio, le funzioni restituiscono un valore boolean che indica l'esito della sua esecuzione

# Creare Funzioni

```
function <nome_funzione>(<parametri>) {  
    <lista di azioni>  
}
```

- Esempio:

```
function quadrato($i) {  
    return $i*$i;  
}
```

- Variabili definite in una funzione sono variabili locali, per accedere a variabili globali si usa l'array \$GLOBALS["name"]
- Le funzioni possono essere definite in qualsiasi punto del codice PHP ed utilizzate anche prima di essere definite.

# Usare le Funzioni

- Per utilizzare una funzione non bisogna fare altro che richiamarla (o invocarla)
- Se esiste ad esempio una funzione `abs` sarà sufficiente usarla in questa maniera:
  - `$a = abs($b);`
  - `$c = 4 * abs(-65);`
- Allo stesso modo si usa una funzione creata da noi:
  - `$f = quadrato($d);`

# Istruzioni: if else

- **Sintassi:**

```
if (condizione) {  
    azione1 da effettuare;  
    azione2;  
}  
else { //se la condizione non e' verificata  
    altre azioni;  
}
```

- **Note:**

- Le parentesi graffe servono per raggruppare una serie di azioni
- La clausola else { } è facoltativa, va usata nel caso ci sia un'alternativa se if non soddisfa la condizione indicata fra le parentesi tonde

- **Esempio:**

```
if ($a==$b){  
    echo "sono uguali"; }  
else{ //se la condizione non è verificata  
    echo "sono diversi";  
}
```



# Elseif

- Sintassi:

```
if ($a==$b) {  
    print "uguali";  
}  
elseif ($a==$c) {  
    print "uguale a c";  
}  
elseif ($a==$d) {  
    print "uguale a d";  
}  
...  
else {  
    print "diversi";}
```

## NOTE:

Si tratta di un'altra istruzione IF all'interno di un IF. Il server controlla se il primo (IF) è vero, se è falso va sul elseif, se è falso pure questo continua con gli ELSEIF fino a quando non trova un'alternativa vera oppure l'istruzione finale ELSE (non obbligatoria)

# Cicli: for

Sintassi:

```
for (espressione iniziale; condizione; aggiornamento) {  
    lista azioni;  
}
```

Esempi:

```
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}  
  
for ($a=0; $a<=3; $a++){  
    echo "ciao $a <br>";  
}
```

**NOTA:** Se la variabile non raggiunge la condizione inserita dentro il ciclo si crea un loop infinito.

# Cicli: While

## Sintassi:

```
while(condizione) {  
    azione1;  
    azione2;  
    //azione per far variare la condizione  
}
```

**NOTE:** Il ciclo while dura fino a quando la condizione è vera. Per far questo dobbiamo necessariamente far variare la condizione all'interno del ciclo: Esempio:

```
$a = 0;  
while($a<=3){  
    print "ciao $a <br>";  
    $a++;  
}
```

In questo caso il ciclo while continua fino a quando \$a non raggiunge il valore 3

# Cicli: Do While

- È' simile al ciclo while MA mentre il ciclo WHILE può non essere eseguito, il ciclo DO WHILE si esegue sempre, almeno per una volta.

Questo perché il ciclo Do While inserisce prima le azioni da fare e dopo la condizione. Il server esegue le prime istruzioni, poi legge la condizione e se è sempre vera esegue nuovamente le istruzioni

- **Sintassi:**

```
do {  
    azione1;  
    azione2;  
    //azione per far variare la condizione  
}  
while(condizione)
```



# Cicli: foreach (1)

- Istruzione foreach serve ad iterare sugli elementi di un array:

```
$arr=array("A", "B", "C");  
foreach($arr as $v) {  
    echo "$v <br>";  
}
```

Analogo di:

```
for($i=0; $i<count($arr); $i++)  
{  
    $v=$arr[$i];  
    echo "$v <br>";  
}
```

- Produce:

A  
B  
C

# Cicli: foreach (2)

- Si puo' sapere anche la chiave associata ad ogni valore:

```
$dati=array("nome" => "mario",  
            "cognome" => "rossi", "telef" => "05534566");  
foreach($dati as $k => $v) {  
    echo "$k: $v <br>";  
}
```

- Produce:

**nome: mario**

**cognome: rossi**

**telef: 05534566**

# Switch

L'istruzione Switch si usa se ci sono più alternative da vagliare e non si vogliono inserire più istruzioni *if* annidate

## Sintassi:

```
switch ($a) {  
    case <espressione>;  
        <lista azioni>  
    ...  
    default:  
        <lista azioni>  
    //entra qui se nessuna condizione è verificata  
}
```

Supponiamo per esempio di inserire una variabile e di dover agire in maniera diversa se questa variabile corrisponde a due valori. Con l'istruzione *if* dovremmo scrivere due *if* annidati, con *switch* ne basta uno

```
switch ($a) {  
    case 'ciao':  
        print "ci vediamo presto";  
        break;  
    case 'addio':  
        print "non torni più?";  
        break;  
    default:  
        print "forse tornerai";  
}
```

# Break

- Questa istruzione termina l'esecuzione di un ciclo. Accetta un argomento opzionale che definisce, nel caso di cicli annidati, il livello del ciclo che è da interrompere.

- Esempio:

```
$i = 0; // Uso dell'argomento opzionale
while (++$i) {
    switch ($i) {
        case 5:
            echo "At 5<br>\n";
            break 1; /* Interrompe solo switch. */
        case 10:
            echo "At 10; quitting<br>\n";
            break 2; /* Interrompe switch e while. */
        default:
            break;
    }
}
```



# Continue

- Questo comando serve per interrompere il ciclo senza uscire, infatti ritorna all'inizio del ciclo e continua il suo lavoro. NON azzera però il valore delle variabili, si usa se uno o più valori non devono influire.
- Come la funzione 'break', accetta un argomento opzionale che definisce, nel caso di cicli annidati, il livello del ciclo che è da interrompere.
- Esempio:

```
for ($a=1; $a<=4; $a++){  
    if ($a==3)  
        continue;  
    else  
        print "ciao $a <br>";  
}
```

- In questo caso otteniamo:

```
ciao 1  
ciao 2  
ciao 4
```

# require(file) / include(file)

- Includono e valutano il file specifico. Sono identiche in ogni senso eccetto per come trattano gli errori: include() produce un Warning mentre require() restituisce un Fatal Error se il file non viene trovato.
- In altre parole, se si vuole che un file mancante fermi l'esecuzione della pagina È NECESSARIO usare require(). Include() non si comporta in questo modo, lo script continuerà nonostante tutto. NOTA: Attenzione al path.

- Esempio:

- FILE 1 (nome file = 'da\_includere.php' )

```
<?php //file da includere,  
echo "Istruzioni contenute nel file da_includere.php </br>";  
$Lesson = "file da_includere incluso</br>";  
?>
```

- FILE 2 (nome file = 'esempio.php' )

```
<?php  
require 'da_includere.php';  
//require('da_includere.php'); //altro metodo di utilizzo di require()  
/*includendo il file ' da_includere.php ' l'interprete php esegue prima tutte le  
istruzioni contenute in tale file e poi continua ad eseguire le istruzioni  
contenute nel file 'esempio.php' */  
echo $Lesson; // adesso posso usare la variabile contenuta nel file  
'da_includere.php'  
?>
```

# isset()

- La funzione `isset()` si usa per sapere se una variabile è stata impostata.

- Esempio:

```
$a=10;
```

```
if(isset($a))
```

```
    echo '$a è impostata';
```

```
if(!isset($b))
```

```
    echo '$b non è impostata';
```

# unset()

- Questa funzione serve per eliminare una variabile

- Esempio:

```
$var_name = 5;  
unset($var_name);  
$colori = array(3 => 'giallo', 'verde', 'blu', 'viola');  
unset($colori[3]); /*elimina solo il primo elemento e NON  
cambia le chiavi, infatti:*/  
echo "<br>$colori[3]"; //non stampa niente  
echo "<br>$colori[4]<br>"; //stampa la stringa 'verde'  
unset($colori); //elimina tutto l'array
```



# sort()/rsort()

- sort() serve per ordinare gli elementi contenuti in un array. NOTA: modifica le chiavi associate ai valori.
- rsort() esegue l'ordinamento inverso
- Esempio:

```
$colori = array(3 => 'giallo', 'verde', 'blu', 'viola', 'rosso');  
sort($colori); //NOTA: riordina anche le chiavi  
//stampa degli elementi ordinati  
foreach ($colori as $k => $c){  
    echo "<br>$k: $c";  
}
```

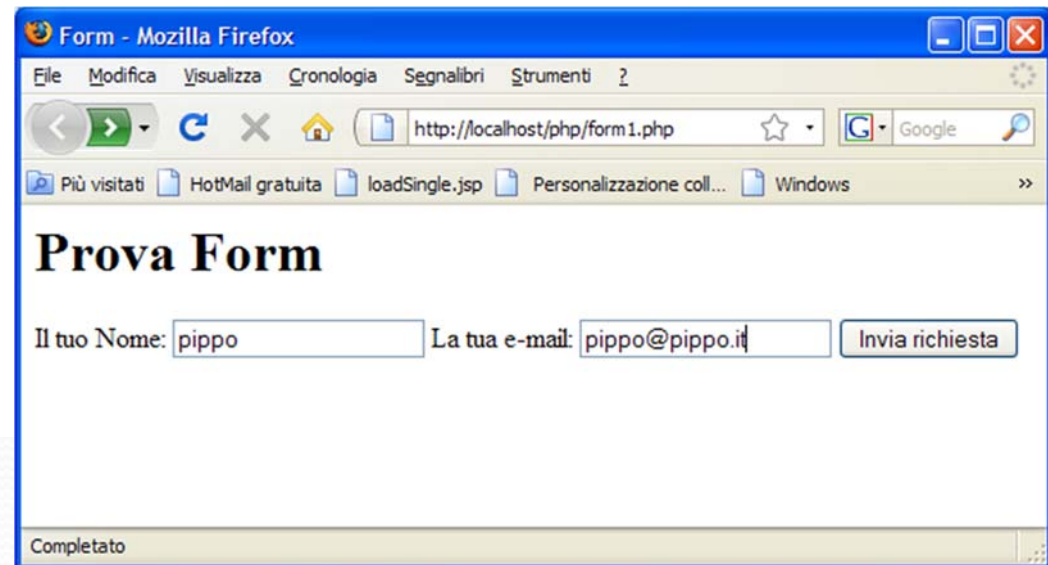
- Produce  
0: blu  
1: giallo  
2: rosso  
3: verde  
4: viola

# HTML FORM

- In HTML esistono tag (FORM, INPUT, TEXTAREA, SELECT,...) che permettono di inviare dati al server.
- I dati sono racchiusi in un form che viene inviato al server in due modi: POST o GET
  - POST: I dati inseriti vengono inviati dal browser al server attraverso la connessione TCP
  - GET: I dati inseriti sono scritti nella URL
- Un form è associato ad una pagina (PHP) che gestirà i dati inviati (attributo **action**), l'output generato da questa nuova pagina sarà la nuova pagina visualizzata sul browser.

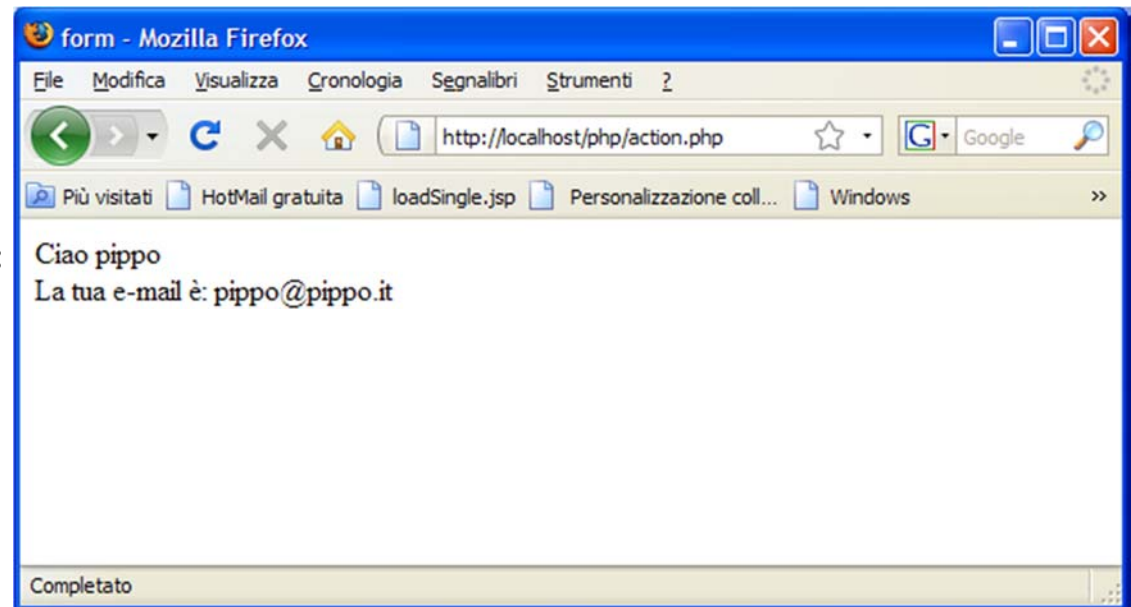
# ESEMPIO (“form.html”)

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-1">
    <title>Form</title>
  </head>
  <body >
    <h1>Prova Form</h1>
    <form action="action.php" method="POST">
      Il tuo Nome: <input type="text" name="name" value="" />
      La tua e-mail: <input type="text" name="email" value ="" />
      <input type="submit">
    </form>
  </body>
</html>
```



# Esempio (“action.php”)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-1">
    <title>form</title>
  </head>
  <body >
    <?php
      echo 'Ciao ';
      echo $_POST["name"];
      echo '<br> La tua e-mail è: ';
      echo $_POST["email"];
    ?>
  </body>
</html>
```





# Esempio 2 (“form.html”)

```
<html>
  <head> <meta http-equiv="Content-Type" content="text/html;
  charset=iso-8859-1">    <title>Form2.it</title>
</head>
<body >
  <h1> Prova Form con tabella per l'allineamento </h1>
  <form action="action.php" method="POST">
    <table align="left">
      <tr>
        <td> Il tuo Nome: </td>
        <td> <input type="text" name="name" value="" /> </td>
      </tr><tr>
        <td> La tua e-mail: </td>
        <td> <input type="text" name="email" value="" /> </td>
      </tr><tr>
        <td> <input type="submit" value="Invia..."> </td>
      </tr>
    </table>
  </form>
</body>
</html>
```



# [...] Esempio 3 ("form.html" )

```
<form action="action.php" method="POST">
  <table>
  ...
  <tr>
    <td> A quale anno di corso sei interessato? </td>
    <td>
      <select name="year">
        <option value="2004">2004</option>
        <option value="2005">2005</option>
        <option value="2006">2006</option>
        <option value="2007">2007</option>
        <option value="2008">2008</option>
      </select>
    </td>
  </tr>
  ...
</table>
</form>
```

[...]

NOTA: nel file action.php si deve usare la variabile \$\_POST["year"]

Form2.it - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://localhost/php/form3.php

Più visitati HotMail gratuita loadSingle.jsp Personalizzazione coll... Windows WindowsMedia >>

## Prova Form con tabella per l'allineamento

Il tuo Nome: Mario Rossi

La tua e-mail: rossi@gmail.com

A quale anno di corso sei interessato? 2006

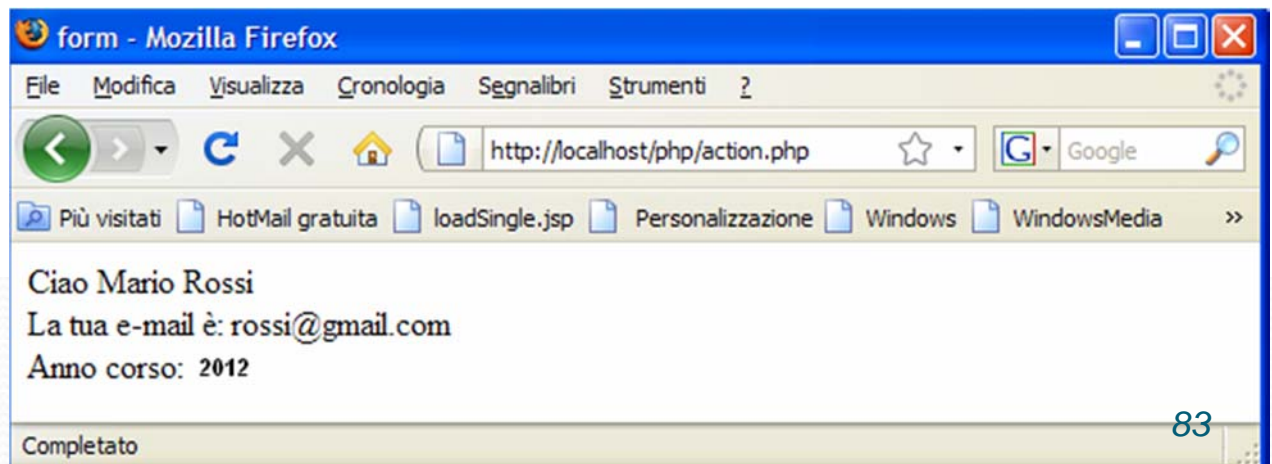
Invia...

2004  
2005  
2006  
2007  
2008

Completato

# Esempio 3 (“action.php”)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-1">
    <title>form</title>
  </head>
  <body >
    <?php
      echo 'Ciao ';
      echo $_POST["name"];
      echo '<br> La tua e-mail è: ';
      echo $_POST["email"];
      echo "<br> Anno corso: $_POST[year]";
    ?>
  </body>
</html>
```





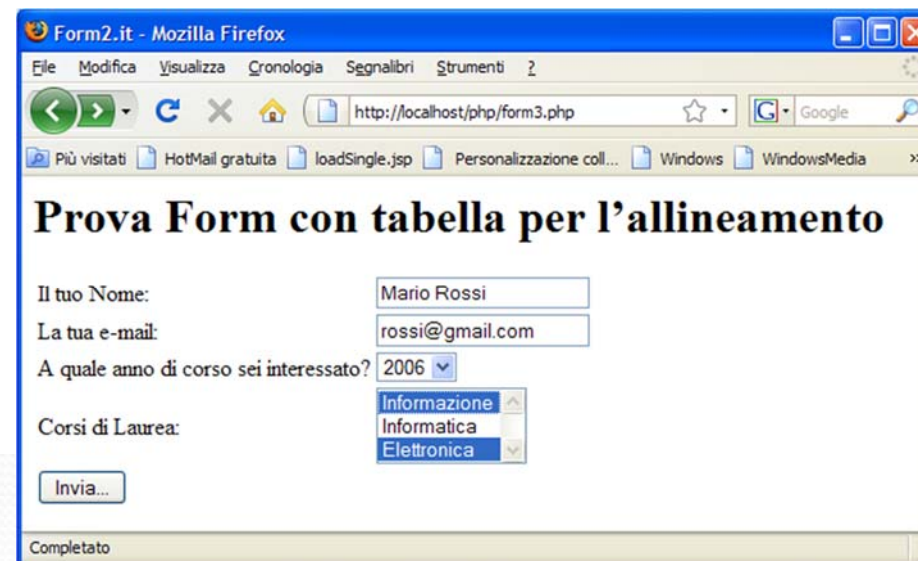
# Esempio 4 (“form.html”)

[...]

```
<form action="action.php" method="POST">
  <table>
    <tr>
      <td>Corsi di Laurea: </td>
      <td>
        <select name="corsi[]" multiple size="5" >
          <option value="Informazione">Informazione</option>
          <option value="Informatica">Informatica</option>
          <option value="Elettronica">Elettronica</option>
        </select>
      </td>
    </tr>
  </table>
</form>
```

[...]

NOTA: nel file action.php si deve usare la variabile `$_POST["corsi"]`



Form2.it - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti 2

http://localhost/php/form3.php

Più visitati HotMail gratuita loadSingle.jsp Personalizzazione coll... Windows WindowsMedia >>

### Prova Form con tabella per l'allineamento

Il tuo Nome: Mario Rossi

La tua e-mail: rossi@gmail.com

A quale anno di corso sei interessato? 2006

Corsi di Laurea: Informazione Informatica Elettronica

Invia...

Completato



# Esempio 4 (“action.php”)

- Ecco come usare la variabile corsi nel file **action.php**:

```
[...]
// un elenco a scelta multipla restituisce un array di
// valori
// se nessun valore è stato selezionato però
// $_POST['corsi'] non sarà settato

if(isset($_POST['corsi']))
    $valori_selezionati = implode($_POST['corsi'], ', ');
else
    $valori_selezionati = 'Nessun valore selezionato';
echo($valori_selezionati);
[...]
```

# Esempio 5 (“form.html” )

[...]

```
<p>campo di testo:<br><input type="text" name="textfield">
</p>
```

```
<p>area di testo:<br> <textarea name="textarea"></textarea>
</p>
```

```
<p><input type="checkbox" name="checkbox" value="checkbox">
      casella di controllo
```

```
</p>
```

```
<p>
```

Fai una delle seguenti scelte: <br>

```
<label> <input type="radio" name="radio" value="1">
      Pulsante di scelta 1
```

```
</label> <br>
```

```
<label> <input type="radio" name="radio" value="2">
      Pulsante di scelta 2
```

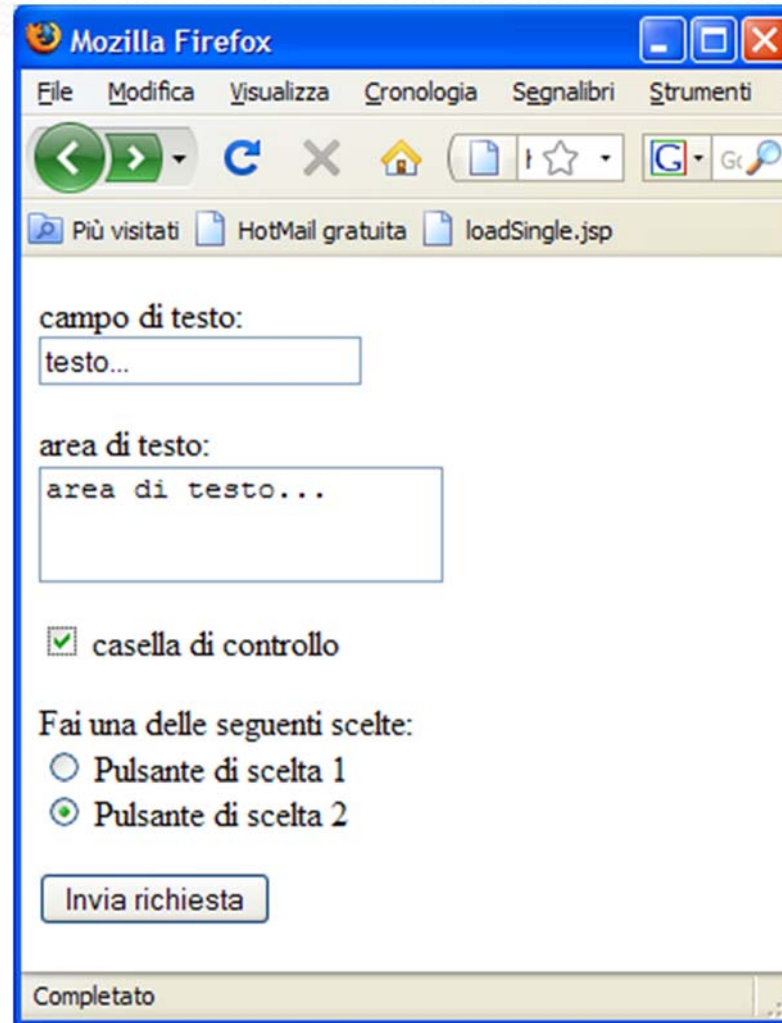
```
</label>
```

```
</p>
```

[...]

NOTA: nel file action.php si devono usare le variabili textfield, textarea, checkbox, radio

# Esempio 5 (“form.html” )



The screenshot shows a Mozilla Firefox browser window with the following elements:

- Menu Bar:** File, Modifica, Visualizza, Cronologia, Segnalibri, Strumenti.
- Navigation Bar:** Back, Forward, Reload, Home, Stop, Address bar (with a search icon), Search engine (Google).
- Bookmarks Bar:** Più visitati, HotMail gratuita, loadSingle.jsp.
- Form Fields:**
  - campo di testo:** A single-line text input field containing the text "testo..."
  - area di testo:** A multi-line text area containing the text "area di testo..."
- Form Elements:**
  - ☒ casella di controllo
  - Fai una delle seguenti scelte:
    - ☐ Pulsante di scelta 1
    - ☒ Pulsante di scelta 2
- Submit Button:** Invia richiesta
- Status Bar:** Completato

# Esempio 5 (“action.php”)

- Ecco come usare la variabili textfield e textarea nel file **action.php**:

```
[...]
// textfield
    // si convertono alcuni caratteri illeciti per l'HTML
    $testo = htmlspecialchars($_POST['textfield']);
    echo('<br>campo di testo: ' . $testo );
    echo('<br>');
// textarea
    $testo = htmlspecialchars($_POST['textarea']);
    //si convertono gli 'a capo' con dei <br>
    $testo = nl2br($testo);
    echo('area di testo: ' . $testo);
    echo('<br>');
[...]
```



# Esempio 5 (“action.php”)

- Ecco come usare le variabili checkbox, radio nel file **action.php**:

```
// checkbox
```

```
// se la casella non è stata selezionata
```

```
// allora $_POST['checkbox'] non sarà settato
```

```
$casella = isset($_POST['checkbox']) ? 'selezionata' :  
'non selezionata';
```

```
echo('casella di controllo: ' . $casella);
```

```
echo('<br>');
```

```
// radio
```

```
// se neanche un bottone è stato selezionato allora
```

```
// $_POST['radio'] non sarà settato
```

```
if(isset($_POST['radio']))
```

```
    $pulsante = 'selezionato il numero ' .
```

```
    $_POST['radio'];
```

```
else
```

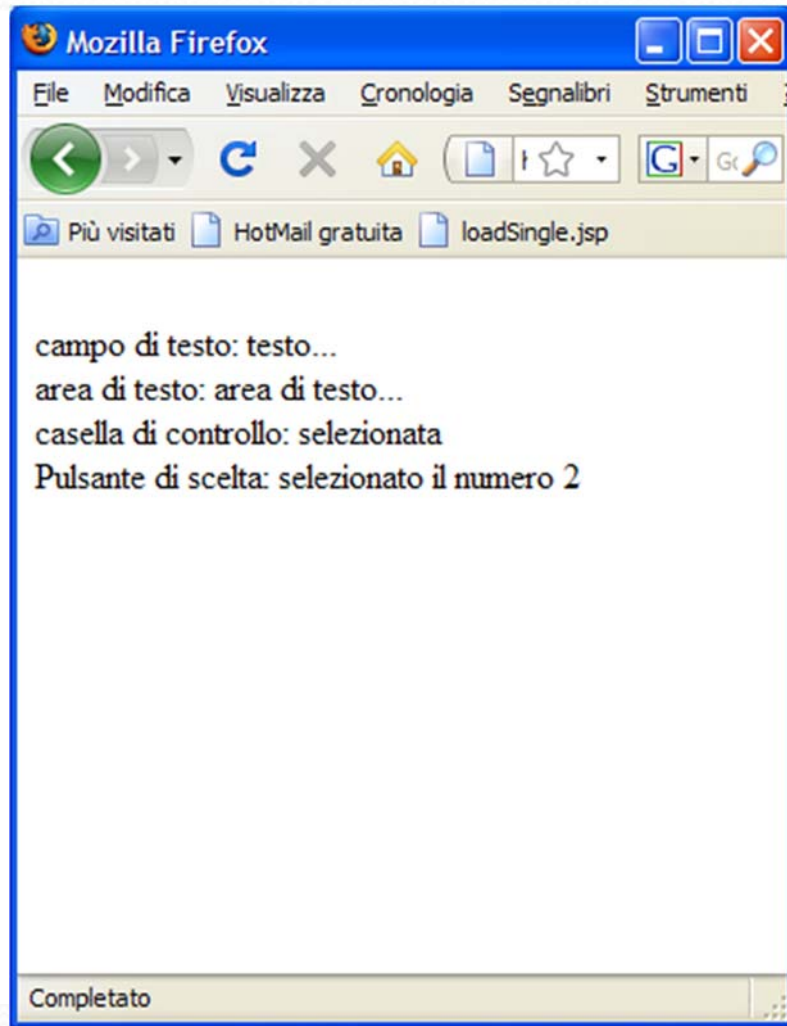
```
    $pulsante = 'nessun pulsante selezionato';
```

```
echo('Pulsante di scelta: ' . $pulsante);
```

```
echo('<br>');
```

[...]

# Esempio 5 (“action.php”)



# Esempio 6 ("form\_GET.html" )

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Form_get.it</title>
  </head>
  <body >
    <h1>
      Prova Form con Metodo get
    </h1>
    <form action="action_get.php" method="get">
      <p>Nome: <input type="text" name="name" /></p>
      <p>email: <input type="text" name="email" /></p>
      <p>
        Che Notizia vuoi?
        <select name="id_notizia">
          <option value="1">Cronaca</option>
          <option value="2">Politica</option>
          <option value="3">Economia</option>
        </select>
      </p>
      <p>
        <input type="submit" />
      </p>
    </form>
  </body> </html>
```



# Esempio 6 (“form\_GET.html”)

Form\_get.it - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://localhost/php/fo

Più visitati HotMail gratuita loadSingle.jsp Personalizzazione coll...

## Prova Form con Metodo get

Nome:

email:

Che Notizia vuoi?

Politica  
Cronaca  
Politica  
Economia

Completato



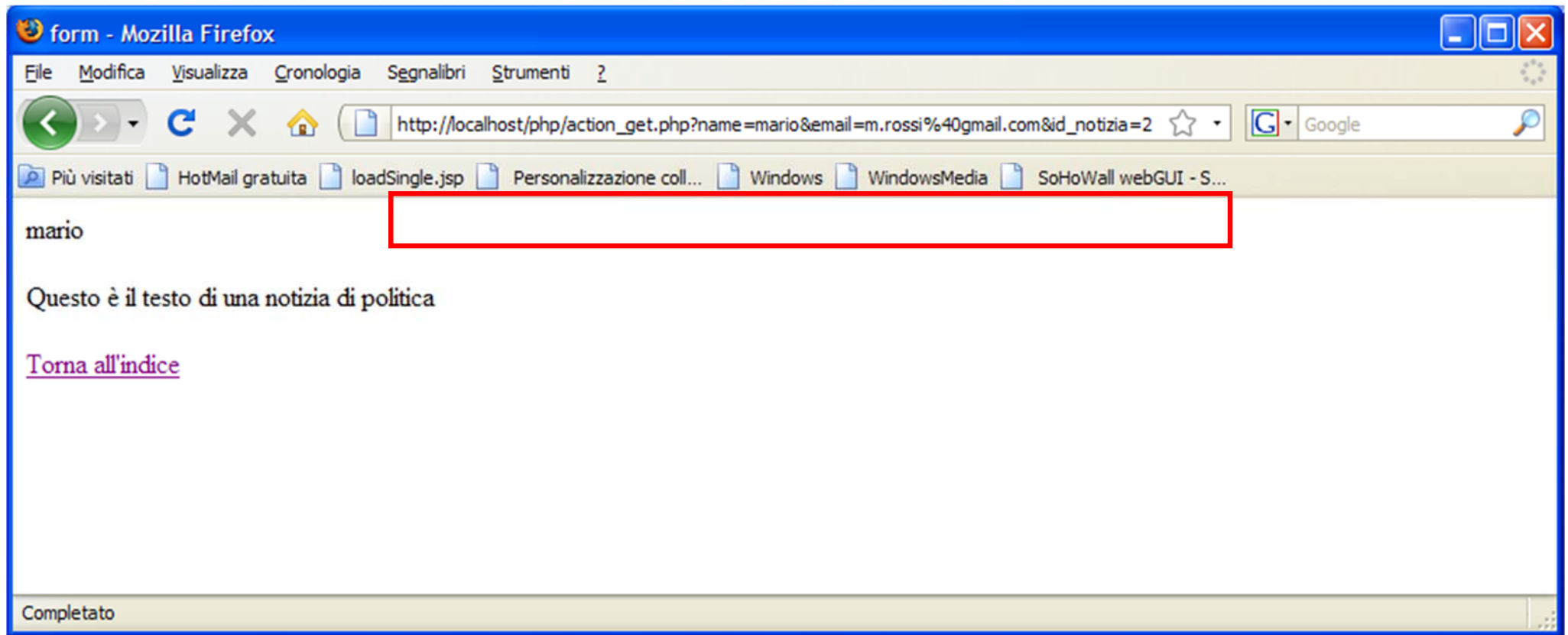
# Esempio 6 (“action\_GET.php”)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <title>form</title>
</head><body >
<?php
  if(!isset($_GET['id_notizia']))
  {
    echo 'Seleziona una notizia da leggere: <br><br>';
    echo '<a href="action_get.php?id_notizia=1">Cronaca</a> <br>';
    echo '<a href="action_get.php?id_notizia=2">Politica</a> <br>';
    echo '<a href="action_get.php?id_notizia=3">Economia</a> <br>';
  }
  else
  {
    $notizie = array(1=>'Questo è il testo di una notizia di cronaca',
                     'Questo è il testo di una notizia di politica',
                     'Questo è il testo di una notizia di economia');

    $id_notizia = $_GET['id_notizia'];

    echo $_GET['name'] . '<br><br>' . $notizie[$id_notizia] . '<br><br>';
    echo '<a href="action_get.php">Torna all\'indice</a>';
  }?> </body> </html>
```

# Esempio 6 (“action\_GET.php”)



# Manipolazione stringhe (1)

- Alcune funzioni:
  - `int strlen($x)`
    - Lunghezza stringa \$x
  - `int strpos($x, $s)`
    - Restituisce la posizione della stringa \$s in \$x o *false* se non esiste
    - `strpos("abcde", "bc") == 1`
  - `string substr($x, $pos)`
    - Restituisce sottostringa di \$x a partire da \$pos ( $\geq 0$ )
    - `substr("abcde", 3) == "de"`
  - `string substr($x, $pos, $len)`
    - Restituisce sottostringa di \$x a partire da \$pos lunga \$len caratteri
    - `substr("abcdefgh", 2, 4) == "cdef"`
  - `string trim($x)`
    - Restituisce la string \$X senza spazi iniziali e finali
  - Continua...

# Manipolazione stringhe (2)

- continua:
  - **array explode(\$delim, \$x)**
    - Restituisce array contenente le parti della stringa \$x delimitate da \$delim
    - `explode(":", "ab:cd:efg:h") == array("ab", "cd", "efg", "h")`
  - **string implode(\$g,\$array)**
    - Restituisce una stringa in cui sono uniti i valori dell'array usando la stringa \$g come separatore tra i valori.
    - `implode(":", array("ab","cd","ef")) == "ab:cd:ef"`
  - **string chr(\$v)**
    - Restituisce stringa formata da un carattere con codice ASCII \$v
    - `Chr(32)==" "`
  - **int ord(\$x)**
    - Restituisce il codice ascii del primo carattere di \$x
    - `Ord(" ")==32`



# Accesso a file

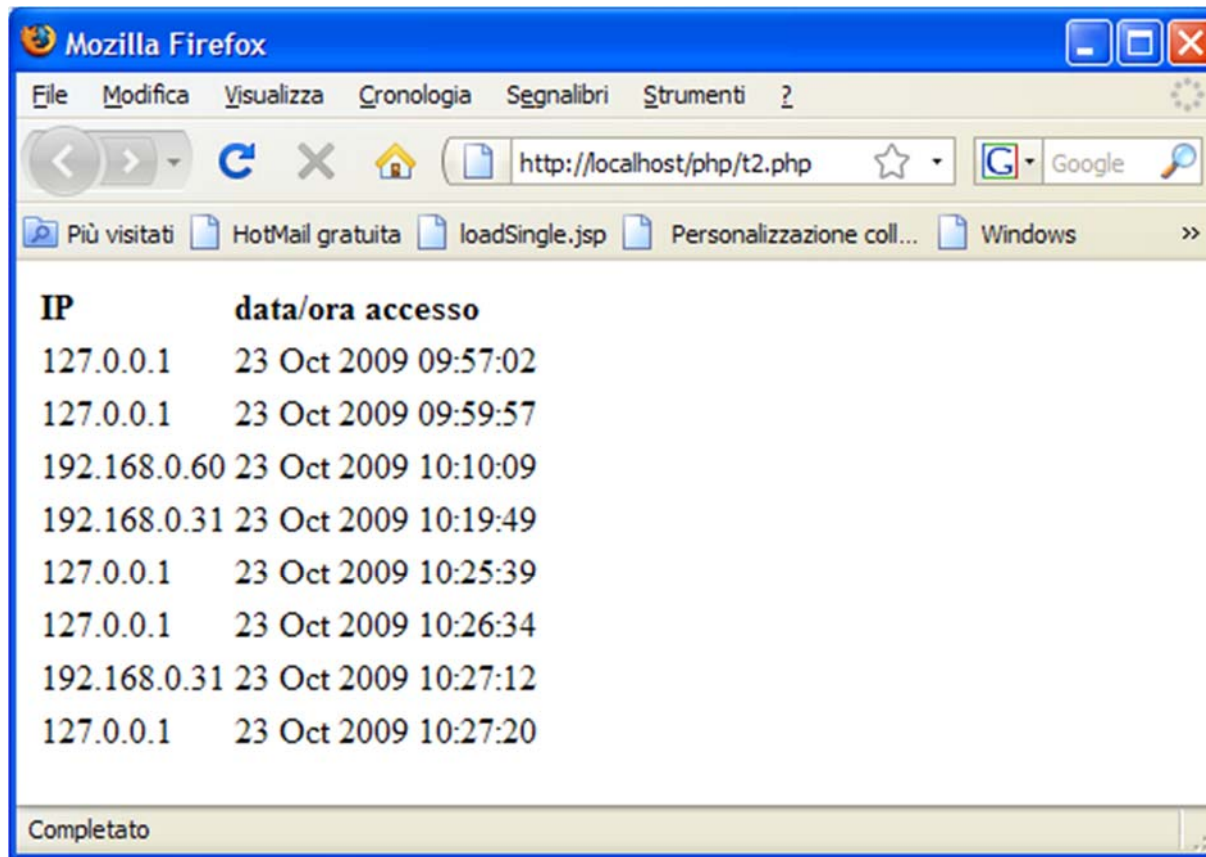
- `array file($url)`
  - Apre il file indicato dalla url e ritorna un array con una stringa per ogni riga
- `resource fopen($url,$mode);`
  - Apre la url indicata nella modalità indicata e restituisce un handle per usarlo o false se non ha potuto aprire il file
  - Modi: 'r', 'r+', 'w', 'w+', 'a', 'a+'
  - Al modo aggiungere 'b' quando si usa un file binario e 't' quando si usa un file testuale, per problemi dei ritorni a capo su windows e linux.
- `fclose($r)`
  - Chiude il file aperto con fopen
- `string fread($r, $len)`
  - Legge \$len caratteri da il file indicato da \$r (ottenuto da fopen) e li restituisce.
- `bool feof($r)`
  - Restituisce true se si è raggiunta la fine del file
- `int fwrite($r, $str)`
  - Scrive la stringa \$str nel file indicato da \$r e restituisce I caratteri effettivamente salvati.

# Esempio

```
<html>
  <body>
    <table>
      <tr><td><b>IP</b></td><td><b>data/ora accesso</b></td></tr>
    <?php
      //apre file in append e se fallisce apertura termina script
      $f=fopen("c:\\PHP\\access.txt", "at") or die("cannot open file");
      $t=time(); //secondi trascorsi da 1/1/1970
      //scrive su file una riga con IP e time, separati da |
      fwrite($f,"$_SERVER[REMOTE_HOST]|$t\n");
      fclose($f);

      //carica in f una array con una riga del file in ogni elemento
      $f=file("c:\\PHP\\access.txt") or die("cannot read file");
      //stampa il contenuto di $f
      foreach($f as $r) {
        //prende da riga ip e time
        list($ip,$t)=explode("|",$r);
        //converte time in una data human readable
        $d=date("d M Y h:i:s", $t);
        echo "<tr><td>$ip</td><td>$d</td></tr>";
      }
    ?>   </table>   </body>   </html>
```

# Esempio



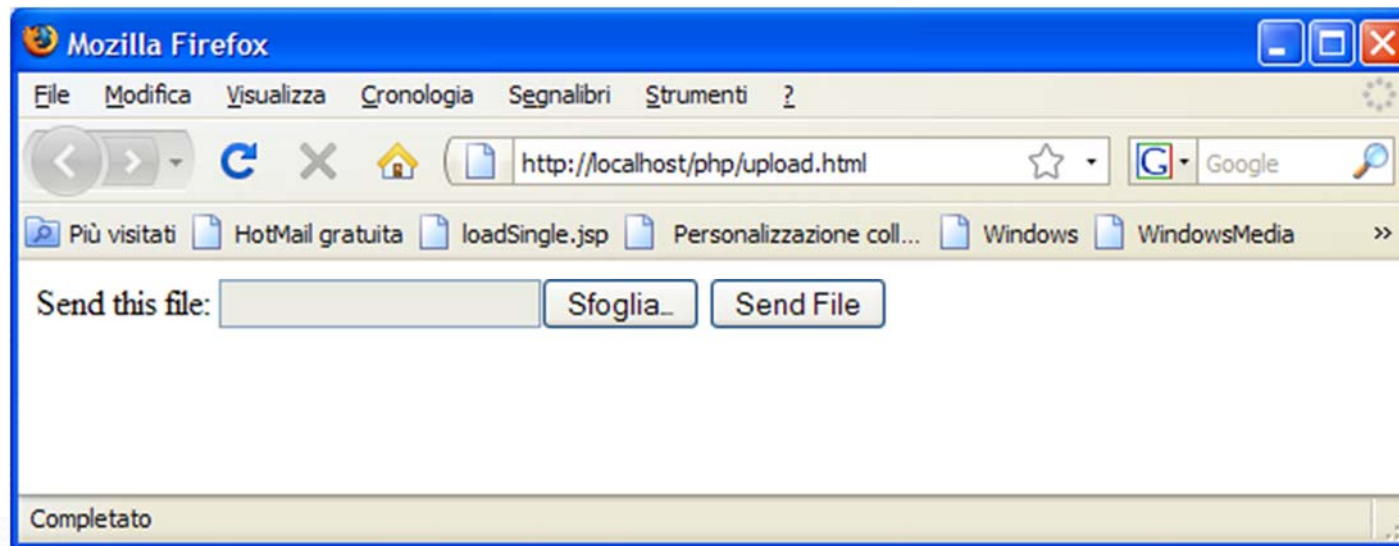


# Form upload file

...

```
<form enctype="multipart/form-data" action="upload.php"
  method="POST">
  <input type="hidden" name="MAX_FILE_SIZE" value="30000" />
  Send this file: <input name="userfile" type="file" />
  <input type="submit" value="Send File" />
</form>
```

...





# Form upload file

- Nello script associato a un form di upload di file si usa l'array **`$_FILES[ nomecampo ]`** per avere informazioni sui file forniti:
  - **`$_FILES['userfile']['name']`** : nome originale del file (sul client)
  - **`$_FILES['userfile']['type']`** : mimetype del file (es: image/png)
  - **`$_FILES['userfile']['size']`** : dimensione del file in bytes
  - **`$_FILES['userfile']['tmp_name']`** : nome del file sul server contenente il file caricato (deve essere copiato da qualche parte)
  - **`$_FILES['userfile']['error']`** : codice d'errore (vedi dopo)

# Form upload file

- Codici errore:
  - ***UPLOAD\_ERR\_OK = 0*** : upload ok
  - ***UPLOAD\_ERR\_INI\_SIZE = 1*** : il file è più grande della dimensione impostata in ***php.ini*** (default 2MB)
  - ***UPLOAD\_ERR\_FORM\_SIZE = 2*** : il file è più grande del valore impostato in MAX\_FILE\_SIZE
  - ***UPLOAD\_ERR\_PARTIAL = 3*** : file parzialmente caricato
  - ***UPLOAD\_ERR\_NO\_FILE = 4*** : file non specificato
  - ...

# Form upload file

- Upload.php:

```
<?php
```

```
$uploadaddir = 'C:/PHP/uploads/';
```

```
$uploadfile = $uploadaddir .
```

```
    basename($_FILES['userfile']['name'])
```

```
echo '<pre>';
```

```
if ($_FILES['userfile']['error']==0 &&  
    move_uploaded_file($_FILES['userfile']['tmp_name'],  
    $uploadfile)) {
```

```
    echo "File valido, caricato correttamente.\n";
```

```
} else {
```

```
    echo "Upload fallito!\n";
```

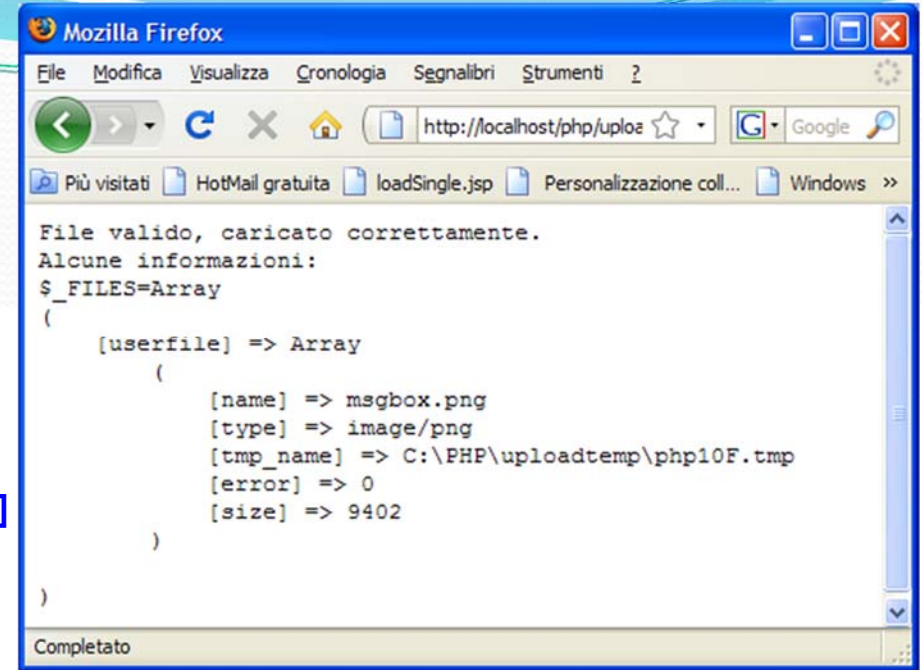
```
}
```

```
echo "Alcune informazioni:\n\$_FILES=";
```

```
print_r($_FILES);
```

```
print "</pre>";
```

```
?>
```



# Form upload file

- In un form si possono specificare anche più input file, in questo caso può essere conveniente indicare di mettere i dati in un array:

```
<form action="file-upload.php" method="post"
  enctype="multipart/form-data">
  Send these files:<br />
  <input name="userfile[]" type="file" /><br />
  <input name="userfile[]" type="file" /><br />
  <input type="submit" value="Send files" />
</form>
```

- In questo modo `$_FILES['userfile']['name']` (e tutti gli altri) contengono array con i dati dei campi input presenti.
  - Es: `$_FILES['userfile']['name']==array('file1.png', 'file2.gif')`



# Sistemi Distribuiti

## Corso di Laurea in Ingegneria

### Programmare per il Web: Il linguaggio PHP

1. Parte I: Introduzione
2. Parte II: PHP
  - CMS
3. Parte III: Approfondimenti

# Content Management System (CMS)

# CMS: origine e definizione

- Un **Content Management System (CMS)** è un Sistema di Gestione dei Contenuti
- E' un Sistema per la creazione, gestione, manipolazione e pubblicazione di Contenuti Digitali su un sito web
- E' uno strumento software installato su un server web studiato per la **realizzazione** e la **gestione** di siti dinamici
- Permette di gestire e organizzare contenuti anche con una conoscenza tecnica minima
- Usando un CMS è possibile ridurre i costi di gestione dei siti internet e intranet disponendo di una elevata flessibilità

# CMS: Vantaggi (1)

- Pubblicazione verso l'utente finale di un insieme di contenuti su più media (PC, device GPRS o UMTS, etc.)
- Aiuta a creare e mantenere siti accessibili: solitamente i CMS seguono le direttive del W3C
  - esempio: possibilità di avere contenuti non dipendenti dal dispositivo sul quale saranno visualizzati. In tale ottica è possibile generare automaticamente codice comprensibile sia dai “lettori braille” sia da device GPRS o UMTS, file di solo testo, etc.



# CMS: Vantaggi (2)

- Esistono CMS specializzati, cioè appositamente progettati per un tipo preciso di contenuti (si pensi ai LCMS, Learning Content Management Systems: Moodle, Docebo, etc.) e CMS generici, che tendono ad essere più flessibili per consentire la pubblicazione di diversi tipi di contenuti
- Un CMS è strutturato in modo che la parte di amministrazione delle risorse possa essere del tutto trasparente all'utente finale. E' costituito da:
  - sezione di amministrazione (back end), serve ad organizzare e gestire la produzione dei contenuti (è accessibile dell'amministratore di sistema e/o dagli sviluppatori del CMS)
  - sezione applicativa (front end): usata dall'utente finale per fruire dei contenuti e delle applicazioni del sito

# CMS: Vantaggi (3)

- Un CMS può essere programmato in vari linguaggi:
  - Prevalentemente: PHP, ASP, NET
- Un CMS permette di aggiornare il sito via web:
  - Qualsiasi utente autorizzato può contribuire alla definizione dei contenuti
  - Questo consente a persone dislocate in sedi diverse, anche lontane fra di loro, di partecipare all'aggiornamento del sito
- Svincola l'amministratore del sito da conoscenze tecniche di programmazione:
  - È possibile costruire e aggiornare un sito dinamico, anche molto grande, senza la necessità di scrivere pagine html e senza conoscere linguaggi di programmazione lato server (come il PHP) o progettare un apposito Data Base
  - L'aspetto esteriore delle pagine (layout o vista) può essere personalizzato scegliendo un foglio di stile tra quelli messi a disposizione dai CMS

# CMS: Vantaggi (4)

- Integra e supporta funzionalità evolute:
  - ad un CMS sono interfacciabili ad esempio:
    - database esterni
    - cataloghi di prodotti
    - sistemi di e-commerce
    - in generale funzionalità web evolute
- Un CMS consente di occuparsi della gestione del sito senza la necessità di ricorrere continuamente ad intermediario (tecnico):
  - molte ditte o enti che, nel corso del tempo, si sono trovate ad avere siti di dimensioni sempre maggiori, hanno anche avuto la necessità di pagare grosse cifre per tenere questi siti aggiornati servendosi di manodopera specializzata



# CMS: Vantaggi (5)

- Dispone di moduli e componenti aggiuntivi:
  - quasi tutti i CMS forniscono dei servizi basilari e dispongono di moduli aggiuntivi che possono essere facilmente integrati al sistema di base per estendere tali funzionalità. (Ad esempio è possibile aggiungere forum, newsletter, sondaggi e altro)
- Notevole velocità nel pubblicare aggiornamenti
- Un CMS permette a molti utenti diversi di pubblicare contenuti direttamente sul sito, ma allo stesso tempo fornisce anche la possibilità di imporre regole per la revisione e l'approvazione dei testi. Permettendo così:
  - consistenza dei dati
  - coerenza dello stile
  - gestione di differenti livelli di sicurezza e di autorizzazioni



# CMS: Funzionamento (1)

- I CMS sono capaci di fornire una serie di servizi così ampia anche grazie alla modalità con cui tali attività vengono progettate e gestite. Si parla infatti di separazione tra **contenuti**, **codice** e **grafica**. Per ciascuna di queste entità si possono individuare i soggetti responsabili:
  - l'azienda o l'ente committente provvede a creare e gestire i **contenuti**
  - il programmatore informatico configura e personalizza la struttura del sistema (**codice**)
  - il grafico web realizza l'interfaccia **grafica** del sito
- Con un CMS viene tenuta distinta l'attività di sviluppo del sito da quella di mantenimento:
  - La prima fase è a cura del programmatore informatico che realizza il progetto (il contenitore con tutti gli automatismi necessari)
  - La seconda fase è a cura degli utilizzatori finali che inseriscono e aggiornano le informazioni del sito (il contenuto)

# CMS: Funzionamento (2)

- I contenuti digitali vengono memorizzati nel server web
- I metadati relativi ai contenuti digitali creati, pubblicati e gestiti nel sito web vengono archiviati in un **Data Base**
- Il DB di un CMS solitamente contiene:
  - la struttura della parte dinamica del sito, che è controllabile dall'utilizzatore
  - i contenuti degli articoli (o i riferimenti), della home e di tutte le parti in gestione all'utilizzatore
  - le regole di visibilità dei contenuti

# CMS: Codice (1)

- E` costituito da un insieme di programmi software (scritti in php o altri linguaggi) che formano il **motore di pubblicazione**:
  - Effettuano il collegamento con il Data Base
  - Inviano in rete il codice html corrispondente ai contenuti presenti nel Data Base
  - Presentano pagine con una struttura fissa (o variabile nel caso di CMS sofisticati) nell'ambito della quale sono visualizzati i contenuti
  - In generale regolano il funzionamento del front-end (ovvero delle pagine web mediante le quali l'operatore gestisce i contenuti del sito):
    - editor testuale o HTML dei contenuti
    - gestione dei livelli di approfondimento
    - gestione dei banner e dei link
    - gestione della visibilità

# CMS: Codice (2)

- Nella maggior parte dei casi, il motore di pubblicazione è ottenuto partendo da una base che è poi personalizzata a seconda dell'utilizzo che ne deve essere fatto:
  - tipo di contenuti gestiti
  - azienda o organizzazione che deve farne uso
  - clienti a cui è destinato
  - etc.



# CMS: Grafica

- La gestione separata in un CMS tra il contenuto digitale e la sua modalità di visualizzazione o presentazione all'utente, permette di modificare agevolmente la veste grafica (layout) del proprio sito web
- Nei CMS è possibile definire dei modelli grafici per la visualizzazione delle pagine web
- Questo permette agli amministratori del sito di:
  - Avere una varietà grafica all'interno del sito scegliendo, fra alcuni template predefiniti, quello che meglio si adatta ai contenuti da pubblicare
  - Personalizzare il proprio sito creando (o richiedendolo agli sviluppatori) propri template

# CMS e contenuti digitali (1)

- Un CMS si occupa in modo automatico di gestire i contenuti digitali. Questo comprende le fasi di:
  - Creazione
  - Acquisizione
  - Conversione
  - Aggregazione
  - Condivisione
  - Archiviazione/Classificazione (metadati, tassonomie, ontologie, etc.)
  - Navigazione
  - Ricerca
  - Etc.
- **Creazione:**
  - un contenuto viene creato ex-novo e caricato nel Data Base del CMS. Il sistema si occupa poi anche della creazione e gestione dei relativi metadati (alcuni in modo automatico, altri da definire in base al contesto)

# CMS e contenuti digitali: Creazione

- Nei CMS è possibile integrare degli editor che consentono di scrivere i contenuti di una pagina o di una sua sezione con le funzionalità standard di un semplice word processor anche senza che l'utente abbia conoscenze di HTML. Si possono effettuare in modo semplice operazioni come:
  - Inserimento di testi
  - Modifica del font e della dimensione del carattere
  - Inserimento di elenchi puntati
  - Inserimento di foto o immagini
  - Inserimento di tabelle
  - Scelta del colore dello sfondo
  - Scelta della posizione di testi e foto
  - ...



## CREATE PAGE

Title: \*



















































Classificazione:














































- Nessuno -
- Genre
- Biography
- Comedy
- Comic
- Drama
- Epic
- Interview
- Life

Select the item you want to insert, you can enter multiple items by holding down the control key

Body:

☒ Show summary in full view

Source                                                   

Format  Font  Size                                              

Scrivo una pagina web usando l'editor (senza bisogno di inserire a mano codice html)

body

Formato input

Lingua:

Italiano

- Gruppi
- Notifiche
- Revision information
- Impostazioni di commento
- File allegati
- Informazioni sull'autore
- Opzioni di pubblicazione

Salva Anteprima

## CONTENUTI

### NUVOLA DELLA KEYWORD

### CLASSIFICATION

### GRUPPI

Thematic Working Groups & Workshops

WG: Proprietà Intellettuale e Modelli di Business

WG: Arti dello Spettacolo, Strumenti Educazionali

WG: Strumenti per le Biblioteche Digitali

Accademia Nazionale di Santa Cecilia

Centre de Documentació Museu de les Arts

Escèniques Institut del Teatre de Barcelona, ITB

Centro Teatro Ateneo, University of Rome La

Per richiedere la creazione di un nuovo gruppo inviare una e-mail all'amministratore. Con: nome del gruppo, nome del coordinatore, breve descrizione del gruppo, lista di eventuali utenti.

### STATISTICS ON YOUR ACTIVITIES

### STATISTICHE SUI TUOI GRUPPI

### E-LEARNING: CORSI

Tutti i corsi  
Tutte le categorie di corsi  
Tutti gli insegnanti  
Le mie attività e-learning

Apple Store, and Windows Phone Market, have a g



Esempio:  
ECLAP  
(European  
Collected  
Library of  
Artistic  
Performance)



# Esempio: ApreToscana

(<http://www.apretoscana.org>)

The screenshot displays the ApreToscana website, which is the Agenzia per la Promozione della Ricerca Europea. The main navigation bar includes links for HOME, CHI SIAMO, SERVIZI, EVENTI, COMUNITÀ, ADERIRE, RICERCHE, and MIO PROFILO. A search bar is located on the right with the text "qualsiasi tipo" and a "deep search" button.

The "CREATE BLOG" section is active, showing a form for creating a new blog post. The "Title" field contains "Evento Aprile 2014". The "Body of the page" field is a rich text editor with a toolbar containing various formatting options like bold, italic, underline, and links. Below the editor, there is a section for "Gruppi" (Groups) with a list of groups: "Miei Gruppi", "Cartario", "Gestione generale", and "ICT e Robotica". A checkbox labeled "Pubblico" is checked, indicating the post will be public.

On the right side of the page, there is a sidebar with several sections: "CONTENUTI" (Most Viewed, Ultimi caricati, Più votati, Mie collezioni pubblicate, Carica un nuovo contenuto), "FROM TWITTER ...", "ORGANIZZA COLLEZIONI PERSONALI", "NUVOLA DELLA KEYWORD", "CLASSIFICAZIONE", "GRUPPI", and "CALENDARIO". The calendar shows the month of March with dates from 1 to 31.

At the bottom of the page, there are logos for "UNIVERSITÀ DEGLI STUDI FIRENZE", "DINFO Dipartimento di Ingegneria dell'Informazione", "DiSiT", "MATCH MAKING multiplace", "APRE Agenzia per la Promozione della Ricerca Europea", "IUF INCUBATORE", and "ITALIA DEGLI INNOVATORI Agenzia per la Diffusione delle Tecnologie per l'Innovazione".

# Esempio: Mobile Medicine

<http://mobmed.axmedis.org>

The screenshot shows the 'Inserisce Story' (Add Story) page of the Mobile Medicine Social Network. The browser window title is 'Inserisce Story - Mobile Medicine Social Network - Mozilla Firefox'. The URL bar shows 'http://mobmed.axmedis.org/index.php?node/add/story'. The page header features the 'Mobile Medicine' logo, the text 'Università degli Studi di Firenze', and a 'powered by axmedis' badge. A 'Welcome root' message with 'Profilo utente' and 'Log out' links is visible. Below the header is a search bar with 'Cerca' and 'any types' and 'any device' dropdowns, and links for 'Ricerca Avanzata', 'GetPlayer', 'Help', and 'Upload'. The main content area is titled 'Inserisce Story' and contains a 'Title' field, a 'Body' text area with placeholder text, and a 'Messaggio di log' field. Below these are links for 'Formato input', 'Impostazioni del menu', 'Impostazioni di commento', 'Informazioni sull'autore', and 'Opzioni di pubblicazione'. At the bottom of the main area are 'Anteprima' and 'Salva e Pubblica' buttons. The right sidebar contains a language selector (English/Italian), a 'Richieste di connessione' section with a list of users and pagination, a 'Cerca utente' section with a search bar and 'Cerca' button, an 'Invita un collega' section with an email input and 'Invita mandando una email' button, and a 'Devel' section with links for 'Devel settings', 'Empty cache', and 'Devel menu'.

# Esempio: editor integrato in un CMS

The screenshot shows a web browser window with the address bar displaying 'http://www.novabmaster.it/ins/padchimento-contenuti-testual\_ICS.html'. The page title is 'Insediamento dei contenuti testuali - Guida PHP-Fusion - CMS - Mr. Webmaster - Mozilla Firefox'. The main content area is titled 'News Estesa:' and contains a large, empty text box. Below the text box is a toolbar with buttons for bold (b), italic (i), underline (u), link, image (img), center, small, small2, and alt. There are also two dropdown menus: '- Seleziona Colore -' and '- Seleziona Immagine -'. Below the toolbar are two date pickers: 'Data Inizio:' and 'Data Fine:', both with '(Opzionale)' below them. The 'Data Inizio:' picker shows '--' for month, '--' for day, and '----' for year, followed by a slash and '0' for hour, '0' for minute, and '00' for seconds. The 'Data Fine:' picker is identical. Below the date pickers is a 'Visibilità:' dropdown menu set to 'Anonimo'. At the bottom are four checkboxes: 'Salva come bozza' (unchecked), 'Rendi questa news importante' (unchecked), 'Taglio Linea Automatico' (checked), 'Abilita Commenti' (checked), and 'Abilita Valutazioni' (checked).



# CMS: esempi di ricerca



**APRE TOSCANA**  
AGENZIA PER LA PROMOZIONE  
DELLA RICERCA EUROPEA

g+1 23

HOME CHI SIAMO ▾ SERVIZI ▾ EVENTI ▾ COMUNITÀ ▾ ADERIRE ▾ RICERCHE ▾ MIO PROFILO

## RICERCA DI PAL/DOMANDE/RICHIESTE

Tutti ▾ Tutte le lingue ▾ Non filtrare per eventi  
Tutti i tipi di attività (PAL) ▾

Ricerca

## RICERCA AVANZATA DI PAL/DOMANDE/RICHIESTE

Seleziona il tipo di contenuto: PAL ▾ Nella lingua: Tutte le lingue  
l'evento: Non filtrare per eventi ▾  
il tipo di attività (PAL): Tutti ▾ il tipo di impresa (PAL): Tutti

Soddisfa una delle condizioni ☐ oppure tutte ☒ nella ricerca

autore  
a.i.o.vene ▾

Aggiunge Rimuovi

Ricerca



HOME ABOUT ▾ MIO PROFILO ▾ CONTENUTI

## RICERCA AVANZATA MULTILINGUA DI OGGETTI

titolo del video video ▾ Italiano ▾ qualsiasi partner ▾

Ricerca

## RICERCA AVANZATA SPECIFICA IN TUTTE LE LINGUE

Oggetti, pagine, forum, commenti

Soddisfa una delle condizioni ☐ oppure tutte ☒ nella ricerca

Title ▾ Contains ▾

Aggiunge Rimuovi

Ricerca Svuota form

Mobile Medicine

Università degli Studi di Firenze



Cerca any types ▾ any device ▾ Ricerca Avanzata

Object search

Metadata

Match any ☐ or all ☒ query criteria

Creator ▾ Contains ▾

Add Remove

Search Clear

Page, Forum, Comment search

Match any

Page ▾

Add Remove

Filter by Taxonomy

None ▾  
Algoritmi problemi e Tecniche di Rianimazione  
Rianimazione Adulto  
Accessi arteriosi  
Accessi venosi centrali e periferici  
ACLS  
BLS  
Complicanze dei devices di gestione delle vie aeree  
Fluidi ed emotrastfusioni

Filter by Group

None ▾  
Master in Clinical Competence in Terapia Intensiva Cardiologica  
Master in Anestesia e Rianimazione  
Broncologia diagnostico interventistica  
Medicina di Emergenza Urgenza

Search Clear



# CMS e Altre Funzionalità

- Sicurezza:
  - Sicurezza a livello di contenuto e servizi (Digital Rights Management, etc.)
  - Gestione di utenti, gruppi e ruoli
  - Controllo del flusso di lavoro (workflow), Notifiche
  - ...
- Lavoro Collaborativo:
  - Gruppi
  - Forum
  - Chat
  - Gallerie fotografiche o di video
  - Condivisione di contenuti e utenti
  - Multi-lingua
  - ...

# Sicurezza: Gestione di utenti, gruppi e ruoli

- Un CMS gestisce gli utilizzatori del sito web grazie ai seguenti concetti:
  - **Utente:** ogni individuo che può accedere al sito web
    - Utente registrato (login e password)
    - Visitatori o utenti anonimi
  - **Ruolo:** ad ogni utente registrato viene assegnato automaticamente un ruolo all'interno del sito (tale ruolo può essere modificato dall'amministratore)
  - **Gruppo:** insiemi di utenti registrati con interessi comuni. Solitamente all'interno di un gruppo operano una serie di utenti con compiti e possibilità di intervento differenziati a seconda del ruolo rivestito nel sito (ad es. Amministratore, utente semplice, etc.)
- L'associazione di questi concetti basilari a determinate categorie di contenuto, pagine o a intere sezioni del portale permette la gestione dei **permessi**, facilita il controllo del flusso delle modifiche, consente l'assegnazione di sezioni del sito alla responsabilità di gruppi specifici, etc.
- Ruoli e permessi regolano in pratica la possibilità di accesso dei diversi utenti ai contenuti e alle informazioni presenti nel sito

# Sicurezza: permessi da associare agli utenti

Permessi  
disponibili per  
gli utenti del  
sito

The screenshot shows the 'Controlla accessi' (Control Access) page in the Mobile Medicine Social Network. The page is titled 'Controlla accessi | Mobile Medicine Social Network - Mozilla Firefox' and the URL is 'http://mobmed.axmedis.org/drupal/?q=admin/user/access'. The page features a search bar, a language selector (English/Italian), and a 'Richieste di connessione' (Connection Requests) section. The main content is a table titled 'Controlla accessi' with columns for 'Permesso' (Permission) and user roles: 'admin', 'anonymous user', 'authenticated user', 'group manager', and 'trusted user'. The table lists various permissions for different modules, such as 'Modulo axmedis', 'Modulo block', 'Modulo comment', 'Modulo devel', 'Modulo favorites', 'Modulo filter', 'Modulo fivestar', 'Modulo forum', 'Modulo friendlist', 'Modulo h18n', 'Modulo invite', 'Modulo locale', 'Modulo menu', and 'Modulo node'. Each permission has a corresponding checkbox in each user role column, indicating whether the permission is granted. A red oval highlights the table, and an arrow points from the text 'Permessi disponibili per gli utenti del sito' to it.

Permesso	admin	anonymous user	authenticated user	group manager	trusted user
<b>Modulo axmedis</b>					
Crea oggetti axmedis	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Modulo block</b>					
administer blocks	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
use PHP for block visibility	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Modulo comment</b>					
Accede ai commenti	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Amministra commenti	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Inserisce commenti	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Inserisce commenti senza approvazione	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Modulo devel</b>					
access devel information	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
execute php code	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
switch users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Modulo favorites</b>					
Select favorites	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Modulo filter</b>					
Amministra filtri	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Modulo fivestar</b>					
rate content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
use PHP for fivestar target	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Modulo forum</b>					
Amministra i forum	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Crea argomenti del forum	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Modifica i propri argomenti del forum	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Modulo friendlist</b>					
Seleziona i colleghi	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Vedi la lista di colleghi	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Modulo h18n</b>					
administer all languages	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Modulo invite</b>					
send invitations	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
send mass invitations	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
track invitations	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
withdraw accepted invitations	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Modulo locale</b>					
Amministra lingue	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Modulo menu</b>					
Amministra i menu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Modulo node</b>					
Accesso al contenuto del sito	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Amministra i titoli di contenuto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



# Sicurezza: Controllo del flusso di lavoro

- Ogni contenuto ha un proprio ciclo di vita all'interno del CMS (**workflow**), in particolare segue un ciclo di approvazione controllato: la messa on-line è l'ultimo step di un percorso di approvazione che permette di pubblicare solo le informazioni ritenute valide e appropriate dall'amministratore del sito o dagli utenti delegati ad approvare i contenuti
- Per accelerare e facilitare il lavoro, spesso vengono predisposte notifiche via e-mail: gli utenti deputati all'approvazione vengono avvertiti quando è necessario il loro intervento
- Le diverse fasi del ciclo di vita sono mantenute separate e svolte spesso da utenti con ruoli differenti: inserimento, visualizzazione anteprima, revisione, pubblicazione finale dei contenuti, mantenimento, modifica, etc.



# Alcuni CMS

- **Drupal** <http://drupal.org/>
- **Moodle** <http://moodle.org/>
- **Docebo**  
<http://web.eccociqua.com/gourl.php?id=2565>
- **Alfresco** <http://www.alfresco.com/it/>
- **Joomla!** <http://www.joomla.org/>
- **Mambo** <http://mambo-foundation.org/>
- **OpenCms** <http://www.opencms.org/en/>
- **Plone** <http://plone.org/>
- **WordPress** <http://wordpress.org/>

# Drupal



# Drupal (1)



- Drupal è un gestore di contenuti (CMS, Content Management System):
  - Distribuito sotto licenza GPL (General Public License)
  - Scritto in linguaggio PHP
  - Installabile sui principali web server (Apache, light httpd o IIS)
  - Si interfaccia con i database Mysql o PostgreSQL
  - Funziona su diversi sistemi operativi, tra cui Windows, Mac OS X, Linux
  - Dotato di una community (sia sviluppatori che utenti) attiva, capace di fornire:
    - Moduli aggiuntivi (rispetto a quelli di base, del Drupal Core)
    - Supporto in caso di bug, modifiche del codice, realizzazione di servizi aggiuntivi per customizzare il proprio sito
  - etc.

# Drupal (2)



- Caratteristiche di Drupal:
  - Modularità
  - Notevole configurabilità, estendibilità e adattabilità
  - Semplice implementazione delle principali caratteristiche tipiche di un sito web orientato alle attività sociali:
    - **Lavoro e attività collaborative:**
      - Gestione di Contenuti ed Utenti (Configurabilità del Profilo utente, etc.)
      - Gestione di gruppi di utenti
      - Condivisione di risorse
      - Uso di Messaging, forum, chat, etc.
    - **Classificazione delle risorse**
      - Possibilità di abilitare moduli per descrivere le risorse (utenti, gruppi, contenuti) attraverso tassonomie e ontologie (XML, RDF, OWL)
    - **Analisi e gestione dei dati** legati alle attività svolte dagli utenti nella piattaforma





Drupal Italia il riferimento italiano per Drupal

Traduzioni Documentazione Forum FAQ  
Contattaci Problemi ricorrenti Chat Team



Cerca nel sito:

Accesso utente

Nome utente: \*

Password: \*



Drupal è un gestore di contenuti (CMS) Open Source realizzato in PHP, che consente di realizzare una grande varietà di siti Web basati sulla logica contenuto/commenti. Le sue doti di robustezza, sicurezza e flessibilità ne stanno decretando la diffusione a livello mondiale.

Drupal Italia è il punto di incontro per favorire la partecipazione di tutta la community di utenti Drupal e gestisce il progetto ufficiale di traduzione e localizzazione italiana. Aiutaci a crescere!

Visita il sito ufficiale di [drupal.org](http://drupal.org)

Versione stabile

**Drupal 6.15**

Translation (.tar.gz)

Translation (.po)

Note di rilascio

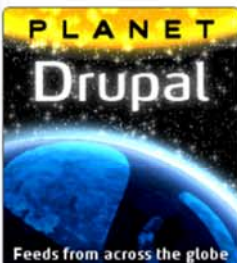
16 Dic 2009 - 22:55

Versione stabile precedente

Drupal 5.21



Documentation Download Support Forum Contribute



Drupal.org is the official website of **Drupal**, an open source **content management platform**.

Equipped with a powerful blend of features, Drupal supports a variety of websites ranging from personal weblogs to large community-driven websites.

► About Drupal  
► Features  
► Hosting  
► Events

► Screenshots  
► Demo  
► Paid services  
► Jobs



Download

Latest release:

Drupal 6.

Drupal 5.21

Contributions:

Modules

Themes

Translations

## Drupal 6.15 and 5.21 released

Gábor Hojtsy - December 16, 2009 - 19:01

News and announcements · Drupal 5.x · Drupal 6.x · Drupal News

Drupal 6.15 and 5.21, maintenance releases which fix issues reported through the bug tracking system, as well as **non critical security vulnerabilities**, are now available for download. Both releases fix other smaller issues as well.

**Upgrading your existing Drupal 5 and 6 sites is strongly recommended.** There are no new features in these releases. For more information about the Drupal 6.x release series, consult the [Drupal 6.0 release announcement](#), more information on the 5.x releases can be found in [Drupal 5.0 release announcement](#).

[Download Drupal 6.15](#)  
[Download Drupal 5.21](#)

» Read more

User login

Username: \*

Password: \*

• [Create new account](#)  
• [Request new password](#)

Contributor links

- Community initiatives
- Queues
  - My issues
  - 513 Pending bugs (D7)
  - 273 Critical issues (D7)
  - 1707 Patch queue (D7)
  - 371 Patches to review (D7)
  - Performance issues (D7)
  - Usability issues (D7)
  - Fields in Core issues (D7)
- Play patch bingo!
- Drupal core

133

## "Drupal: Creating and Editing Custom Themes" training video series from Lynda.com

Chris Charlton - January 7, 2010 - 17:45

News and announcements · Drupal 6.x

New training to help you learn how to build Drupal themes is available online from Lynda.com. *Drupal: Creating and Editing Custom Themes* shows how a Drupal theme can alter the look and feel of a web site. Drupal evangelist and author Chris Charlton introduces the core themes that ship with Drupal, how to install and administer new themes, change color schemes, and add sub-themes. Also demonstrated is how to build a custom theme from scratch, what templates integrate HTML, CSS, and JavaScript; and how to theme specific nodes, pages, and blocks by building a theme step-by-step for a

» 10 comments · Read more



# Drupal: principi di base (1)

- Drupal è una applicazione Web: i contenuti vengono inseriti e visualizzati attraverso un Web Browser (Internet Explorer, Mozilla Firefox, Opera etc.)
- E' possibile realizzare siti Web pubblici o locali (intranet)
- I componenti richiesti per l'installazione di Drupal sono:
  - **Web server:** un server in grado di pubblicare pagine Web. Il Web server predefinito è Apache, ma anche IIS è previsto. La maggior parte degli Internet Provider mette a disposizione l'uno o l'altro Web server. Per realizzare un'installazione locale, è invece necessario installare localmente un Web server
  - **Interprete PHP:** Drupal è un insieme di moduli di codice PHP: le pagine Web vengono generate dinamicamente in base alle richieste ricevute, a partire dal codice PHP. Il sistema che converte il codice PHP in pagine Web fruibili dall'utente è chiamato *interprete PHP*. Per un'installazione Web, verificare che il proprio provider supporti PHP
  - **Database SQL:** i dati di configurazione e i contenuti inseriti dagli utenti vengono archiviati in un database SQL. Il database predefinito è MySQL, ma praticamente qualsiasi database supportato da PHP è utilizzabile



# Drupal: principi di base (2)

- Consente agli utenti di registrarsi e autenticarsi in modo da tenere traccia di chi è autore di ogni singolo contenuto
- Permette agli amministratori del sito di consentire livelli di accesso differenziati a seconda dei ruoli (utente, moderatore, amministratore, etc.)
- Attivando e configurando i singoli moduli, un amministratore può customizzare il sito. In particolare è possibile attivare moduli (e quindi servizi) legati a:
  - Gestione dei Contenuti
  - Weblog
  - Comunità basate sulla discussione
  - Collaborazione
  - etc.

# Drupal: principi di base (3)

- **Gestione dei Contenuti:**

- Attraverso un'interfaccia semplice e fruibile via browser i membri possono pubblicare contenuti nei diversi moduli disponibili: storie, blog, sondaggi, immagini, forum, scaricamenti, etc.
- Gli amministratori possono scegliere tra diversi template di temi o crearne uno essi stessi per dare al sito un look ad feel unico
- Il sistema di classificazione flessibile consente una classificazione gerarchica, indici incrociati di messaggi e diversi insiemi di categorie per la maggior parte dei tipi di contenuti
- L'accesso al contenuto è controllato attraverso ruoli di permessi utenti definiti dall'amministratore
- Le pagine del sito possono mostrare messaggi per tipo di modulo o per categorizzazione del contenuto, con fonti RSS separate disponibili per ogni tipo pubblicato. Gli utenti possono anche cercare per parola chiave all'interno dell'intero sito.



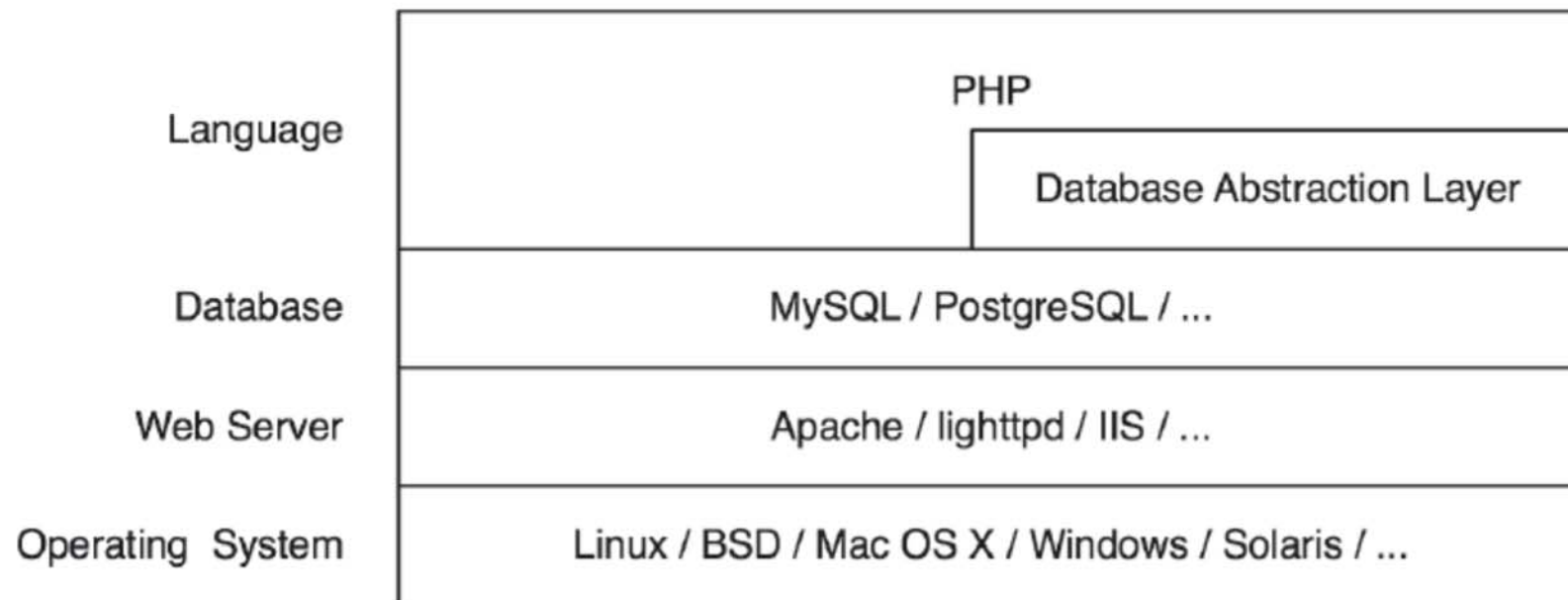
# Drupal: principi di base (4)

- **Weblog:**
  - Una singola installazione può essere configurata come sito weblog personale individuale o meno
  - Drupal supporta le API Blogger, fornisce fonti RSS per ogni blog individuale e può essere impostato per lanciare un ping alle directory di weblog come blo.gs e weblog.com quando un nuovo contenuto è inserito nella home page
- **Comunità basate sulla discussione. Un sito Drupal può:**
  - Fare uso di uno o più forum di discussione tradizionali
  - Attivare i commenti, collegati alla maggior parte dei tipi di contenuto, è un metodo semplice per i membri per discutere, inviare messaggi, etc.
  - Gli amministratori possono determinare se il contenuto e i commenti possono essere inseriti senza approvazione, con l'approvazione di un amministratore o per moderazione della comunità
  - Grazie all'aggregatore di notizie interno, le comunità possono registrarsi su altri siti e discutere contenuti provenienti da essi
- **Collaborazione:**
  - E' possibile attivare servizi orientati al controllo delle versioni, rendendo semplice per un gruppo di utenti creare, revisionare e aggiornare documentazione o altri tipi di testo

# Drupal: installazione (1)

- Se si vuole usare **Drupal in locale**, l'installazione dei tre componenti (Web server, Interprete PHP, Database SQL) può risultare abbastanza complessa, per cui è consigliato scaricare XAMPP: un pacchetto integrato realizzato appositamente per lo scopo. È un software gratuito e disponibile per le più diffuse piattaforme (Windows, Mac OS X, Linux)
- E' necessario scaricare il pacchetto con tutti i file dal sito ufficiale: un archivio ZIP, dal quale si può estrarre i file e copiarli in una cartella sul nostro PC (da determinare in base alla XAMPP usata)
- Una volta caricati tutti i file, si seleziona sul browser il sito (es. <http://www.nomedominio.com>) e verrà mostrata la pagina iniziale dell'installazione

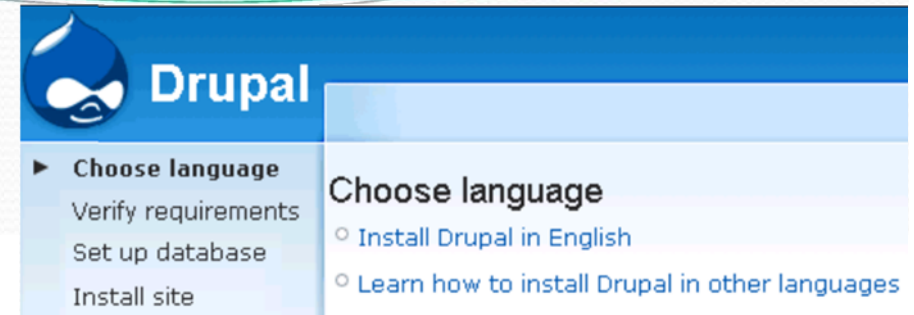
# Drupal's technology stack





# Drupal: installazione (2)

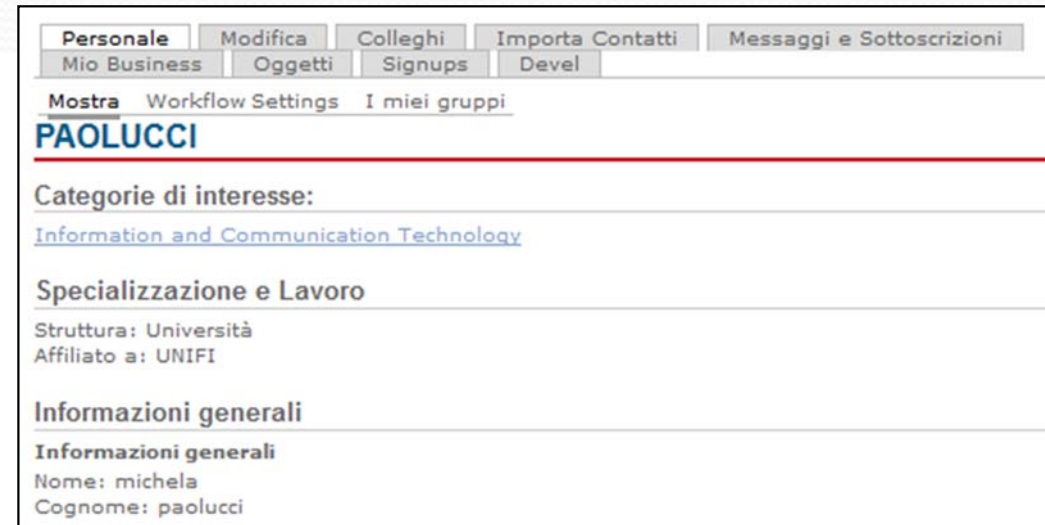
- **Selezione Lingua:** Per installare Drupal in italiano potrebbe essere necessario importare un file con la traduzione (<http://drupal.org/project/it>)
- **Database:**
  - Solitamente si usa mysql
- **Configurazione del sito:**
  - Nome del sito
  - e-mail





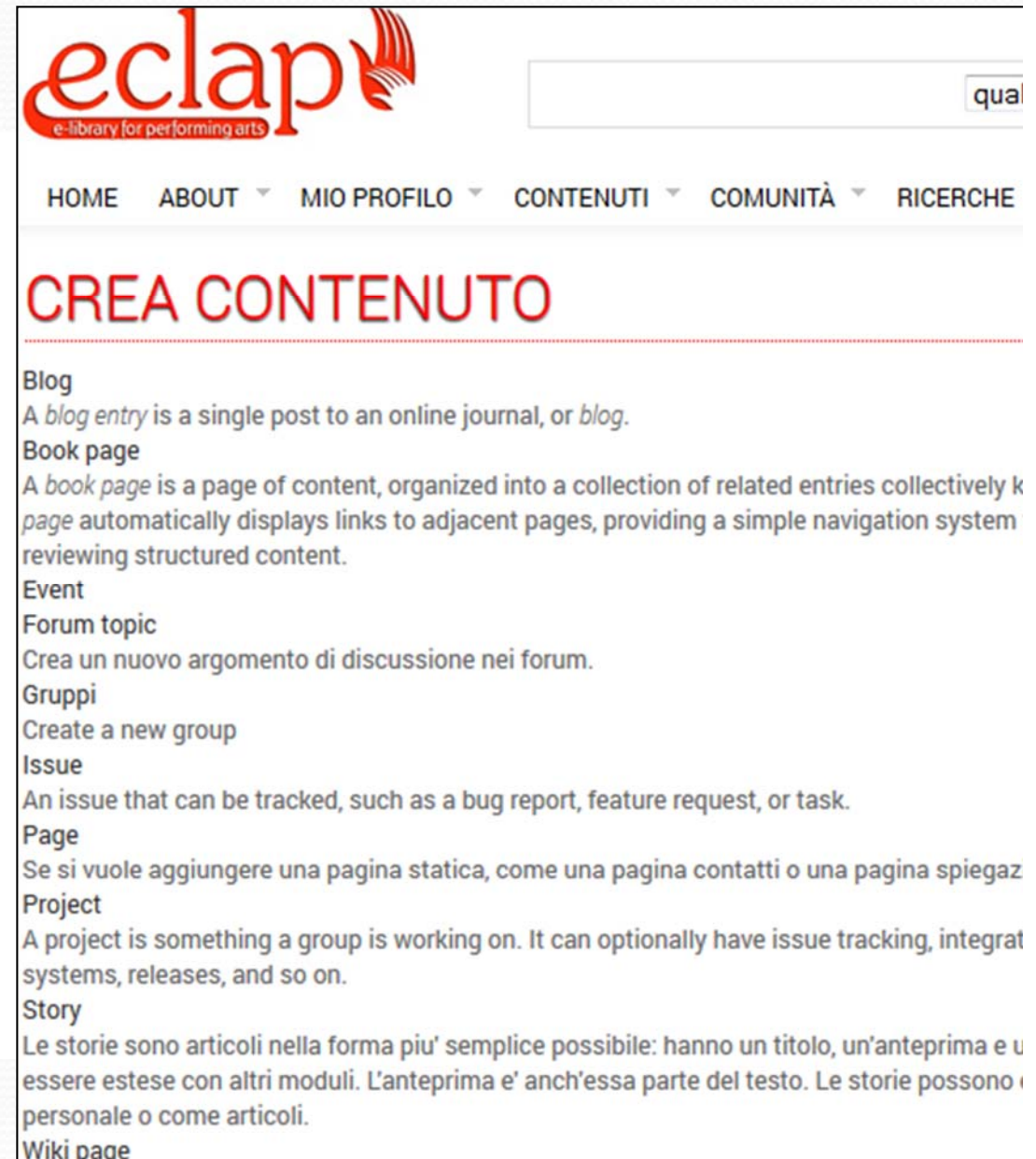
# Drupal: installazione (3)

- Dopo avere effettuato l'installazione, si inseriscono **login e password** di amministratore. Si visualizzano dei link accessibili solo all'amministratore, tra cui:
  - **Profilo utente:** si visualizzano alcune informazioni relative all'utente root (o agli altri utenti se sono stati creati), inoltre è possibile cambiare le impostazioni principali dell'account (es: nome utente, l'indirizzo e-mail, la password, lo stato, il fuso orario e il linguaggio predefinito, etc.)



# Drupal: installazione (4)

- Dopo avere effettuato l'installazione, si inseriscono **login e password** di amministratore. Si visualizzano dei link accessibili solo all'amministratore, tra cui:
  - **Crea contenuto:** L'installazione standard prevede due sole categorie di contenuto, ma è possibile aggiungerne di nuove (utilizzando moduli Drupal aggiuntivi). Le tipologie di base sono:
    - Pagina web (di default è possibile aggiungere dei commenti)
    - Storia (si comporta come un post di un blog, a cui è assegnata una data ed a cui è possibile aggiungere dei commenti)



The screenshot shows the 'eclap' website interface. At the top, there is a logo for 'eclap e-library for performing arts' and a search bar. Below the logo, there is a navigation menu with links: HOME, ABOUT, MIO PROFILO, CONTENUTI, COMUNITÀ, and RICERCHE. The main heading is 'CREA CONTENUTO'. Below this, there is a list of content types with their descriptions:

- Blog**  
A *blog entry* is a single post to an online journal, or *blog*.
- Book page**  
A *book page* is a page of content, organized into a collection of related entries collectively k page automatically displays links to adjacent pages, providing a simple navigation system reviewing structured content.
- Event**
- Forum topic**  
Crea un nuovo argomento di discussione nei forum.
- Gruppi**  
Create a new group
- Issue**  
An issue that can be tracked, such as a bug report, feature request, or task.
- Page**  
Se si vuole aggiungere una pagina statica, come una pagina contatti o una pagina spiegaz
- Project**  
A project is something a group is working on. It can optionally have issue tracking, integrat systems, releases, and so on.
- Story**  
Le storie sono articoli nella forma piu' semplice possibile: hanno un titolo, un'anteprima e u essere estese con altri moduli. L'anteprima e' anch'essa parte del testo. Le storie possono i personale o come articoli.
- Wiki page**

# Drupal: Concetto di nodo

- La gestione del contenuto di Drupal è molto particolare, è possibile inserire una grande varietà di contenuti di diverso tipo che hanno tutti una caratteristica comune, ovvero sono dei **nodi**:
  - Pagine
  - Utenti
  - Gruppi di utenti
  - Nuove tipologia di contenuti
  - Etc.
- Tutti i tipi di contenuto hanno attributi come: autore, data di creazione, titolo, etc.
- Ogni contenuto può essere associato ad una categoria e ricevere commenti da parte degli utenti, avere associati degli attachments, essere impostato in modo da tenere traccia delle revisioni, etc.
- Drupal tratta ogni contenuto prima come un nodo, per semplificarne la gestione, quindi entra nei dettagli e può così separare un articolo da un post del forum o da una pagina, etc.



# Drupal: amministrazione (1)

- Dal menù di amministrazione è possibile configurare il CMS scegliendo tra due modalità di visualizzazione:
  - per funzionalità: visualizza la stessa struttura navigazionale presentata nella sidebar, raggruppando le funzionalità offerte in base alla tipologia di operazione compiuta da ciascuna opzione
  - per modulo: raggruppa le funzionalità in base al modulo che le offre. Quando viene aggiunto un modulo al sistema vengono incrementate le opzioni e viene aggiunto un nuovo gruppo di configurazione



The screenshot shows the Drupal administration interface for the website <http://www.disit.dinfo.unifi.it>. The header includes the logo for the Distributed Systems and Internet Technologies Lab (D<sup>ISI</sup>T) and the Distributed Data Intelligence and Technologies Lab (D<sup>IDI</sup>T), both part of the Department of Information Engineering (DINFO) at the University of Florence. A navigation menu includes links for HOME, ABOUT, RESEARCH, INNOVATION, CORSI E TESI, and COME FARE. Below the menu, there are two tabs: "Per funzionalità" (selected) and "Per modulo". The main content area is titled "AMMINISTRA" and lists various administrative functions under the heading "Nasconde descrizioni".

Gestione contenuti	Struttura del sito
<b>Gestisce i contenuti del sito.</b> <a href="#">Commenti</a> Elenca e modifica i commenti del sito e la coda di moderazione dei commenti. <a href="#">Contenuto</a> Vede, modifica, e cancella i contenuti del sito. <a href="#">Impostazioni di invio</a> Controlla il comportamento del contenuto, ossia la lunghezza delle anteprime, richieste di visualizzazione anteprima prima di inserire il contenuto e il numero di contenuti da inserire sulla prima pagina. <a href="#">Libri</a> Manage your site's book outlines. <a href="#">Moderation</a> Thread moderation queue <a href="#">Pubblicazione RSS</a> Configura il numero di voci per ogni feed e se i feeds dovranno essere costituiti da titolo, sommario o testo completo. <a href="#">Signup administration</a> View all signup-enabled posts, and open or close signups on them.	<b>Controlla l'aspetto del Vo</b> <a href="#">Blocchi</a> Configura quale contenut apparire nelle barre later aree <a href="#">Menù</a> Controlla il menu' di navi primari e secondari, così riorganizza gli oggetti del <a href="#">Moduli</a> Attiva o disattiva moduli per il sito. <a href="#">Pagine</a> Add, edit and remove ov pages and user defined p system. <a href="#">Pannelli</a> Amministra le voci relativ <a href="#">Tem</a> Cambia il tema usato dal utenti di impostarne uno. <a href="#">Traduci interfaccia utente</a> Translate the built in inte



# Drupal: amministrazione (2)

- Configurazioni di base:
  - Gestione dei contenuti
  - Struttura del sito: per impostare le configurazioni di personalizzazione del comportamento e delle funzionalità del sito

## Gestione contenuti

### [Commenti](#)

Elenca e modifica i commenti del sito e la coda di moderazione dei commenti.

### [Contenuto](#)

Vede, modifica, e cancella i contenuti del sito.

### [Impostazioni di invio](#)

Controlla il comportamento anteprima prima di inserire

### [Pubblicazione RSS](#)

Configura il numero di voci testo completo.

### [Tipi di contenuto](#)

Gestisce i contenuti per tipo

## Struttura del sito

### [Blocchi](#)

Configura quale contenuto del blocco debba apparire nelle barre laterali del sito e in altre aree

### [Menu'](#)

Controlla il menu' di navigazione del sito, i link primari e secondari, così come rinomina e riorganizza gli oggetti del menu'.

### [Moduli](#)

Attiva o disattiva moduli aggiuntivi (add-on) per il sito.

### [Temi](#)

Cambia il tema usato dal sito o permette agli utenti di impostarne uno.

### [Visualizzazioni](#)

Le visualizzazioni sono liste personalizzate di contenuti presenti nel sistema; sono altamente configurabili e assicurano il controllo su come le liste di contenuti vengono presentate.

- ## USER SETTINGS

---

### User registration settings

**Public registrations:**

☐ Only site administrators can create new user accounts.  
☐ Visitors can create accounts and no administrator approval is required.  
☐ Visitors can create accounts but administrator approval is required.  
☒ New user registration by invitation only.

☒ Require e-mail verification when a visitor creates an account

If this box is checked, new users will be required to validate their e-mail address prior to logging into the site, and will be assigned a system-generated password. With it unchecked, users will be logged in immediately upon registering, and may select their own passwords during registration.

---

### User registration guidelines

This text is displayed at the top of the registration page.

---

### User e-mail settings

Drupal sends emails whenever new users are created or perform actions. Using a simple set of content replacements, you can customize the messages.

▼ [Welcome, new user created](#)

Customize welcome e-mail message. Replacements are: !username, !site, !password, !url.

**Subject:**

An administrator created an account for you.

**Body of the page:**

```
<html>
<head>
<meta content="text/html" charset="utf-8" />
<meta name="viewport" content="width=device-width, height=device-height, scale=0.1, maximum-scale=1" />
<body>
```

---

### Signatures

**Signature support:**

☒ Off  
☐ Active

---

### Pictures

**Picture support:**

☐ Off  
☒ Active

**Picture image path:**

Subdirectory in the directory files/ where pictures will be stored.

**Default picture:**

URL of picture to be displayed for users with no custom picture selected. Leave blank for none.

**Picture maximum dimensions:**

Maximum dimensions for pictures, in pixels.

**Picture maximum file size:**

Maximum file size for pictures, in kB.

**Picture guidelines:**

This text is displayed at the picture upload form in addition to the default guidelines. It's useful for helping or instructing your users.

---

### Administrator role

**Administrator role:**

This role will be automatically assigned new permissions whenever a module is enabled.



# Drupal: amministrazione (4)

- **Gestione utente:** serve per gestire gli utenti iscritti al sito. Consente di:
  - **Permessi:** consente di impostare quali sono le operazioni consentite a ciascun ruolo di utenti. I ruoli predefiniti sono: utente anonimo ed utente registrato, ma è possibile impostarne di nuovi (ad esempio moderatore)

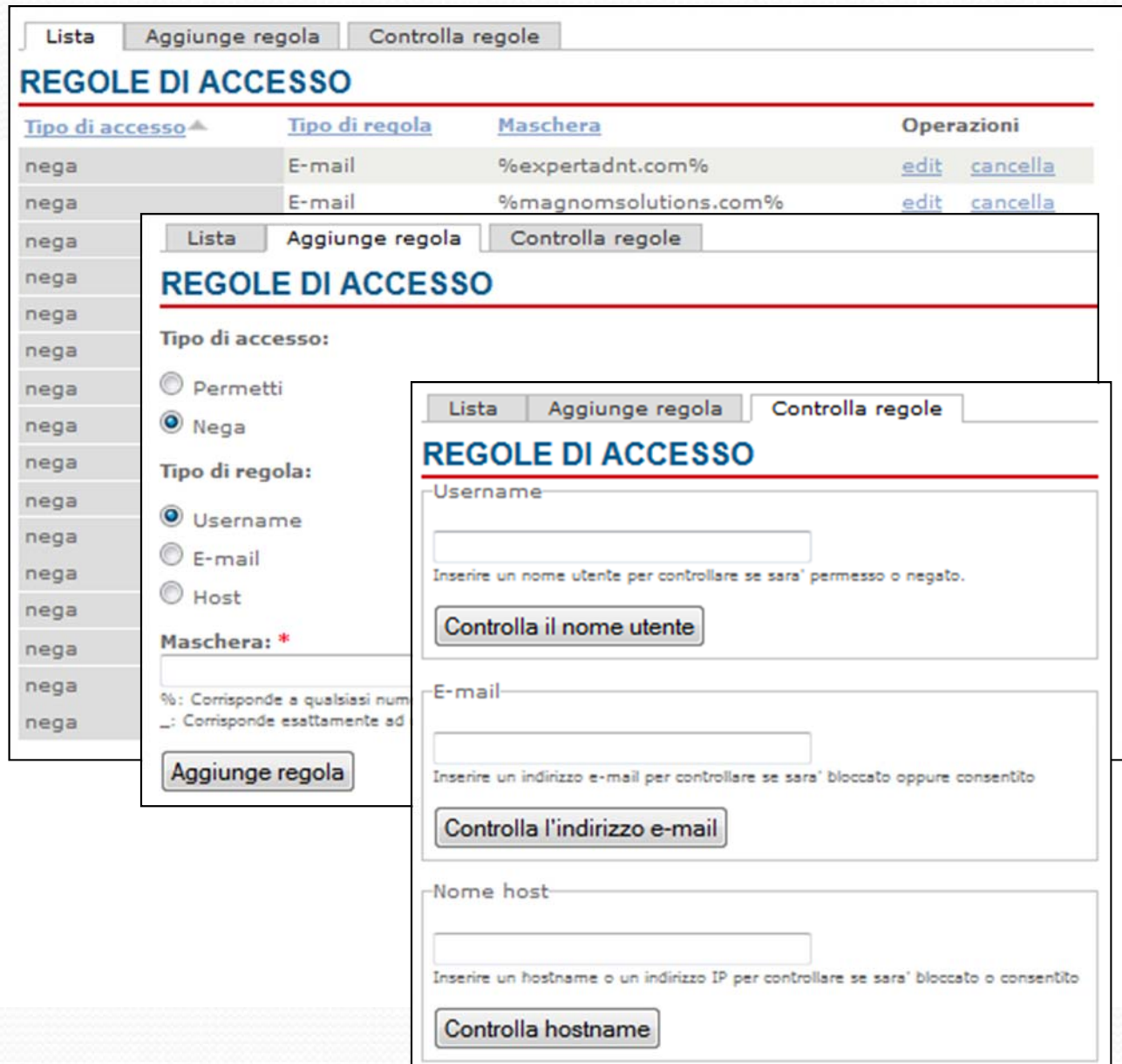
## Controllo accessi

Permesso	admin	anonymous user	authenticated user	group manager	trusted user
<b>Modulo comment</b>					
Accede ai commenti	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Amministra commenti	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Inserisce commenti	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Inserisce commenti senza approvazione	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Modulo forum</b>					
Amministra i forum	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Crea argomenti del forum	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Modifica i propri argomenti del forum	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Modulo node</b>					
Accesso al contenuto del sito	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Amministra i tipi di contenuto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Amministra i nodi	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
create groups content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
create page content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
create story content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
edit groups content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
edit own groups content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
edit own page content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
edit own story content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
edit page content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
edit story content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ripristina revisioni	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Guarda revisioni	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>Modulo user</b>					
Accede ai profili utente	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Amministra il controllo accessi	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Amministra utenti	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cambia il suo nome utente	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



# Drupal: amministrazione (5)

- **Gestione utente:** serve per gestire gli utenti iscritti al sito. Consente di:
  - **Regole di accesso:** permette di definire alcune regole per vietare o consentire la registrazione ad alcuni indirizzi e-mail o nomi utente attraverso l'uso di espressioni regolari



The screenshot displays the 'REGOLE DI ACCESSO' (Access Rules) administration page in Drupal. It features a table of existing rules and a form to add a new rule.

Tipo di accesso	Tipo di regola	Maschera	Operazioni
nega	E-mail	%expertadnt.com%	<a href="#">edit</a> <a href="#">cancella</a>
nega	E-mail	%magnomsolutions.com%	<a href="#">edit</a> <a href="#">cancella</a>
nega			
nega			
nega			
nega			
nega			
nega			
nega			
nega			
nega			
nega			
nega			
nega			

**REGOLE DI ACCESSO**

Tipo di accesso:

☐ Permetti

☒ Nega

Tipo di regola:

☒ Username

☐ E-mail

☐ Host

Maschera: \*

%: Corrisponde a qualsiasi num

\_ : Corrisponde esattamente ad

**Username**

Inserire un nome utente per controllare se sara' permesso o negato.

**Controlla il nome utente**

**E-mail**

Inserire un indirizzo e-mail per controllare se sara' bloccato oppure consentito

**Controlla l'indirizzo e-mail**

**Nome host**

Inserire un hostname o un indirizzo IP per controllare se sara' bloccato o consentito

**Controlla hostname**

# Drupal: amministrazione (6)

- **Gestione utente:** serve per gestire gli utenti iscritti al sito. Consente di:
  - **Utenti:** è il pannello dove vengono elencati tutti gli utenti iscritti e dove è possibile registrarne di nuovi

Lista Aggiunge utente

## UTENTI

Informazioni sul profilo utente

Username: \*

Sgli spazi sono permessi; altri simboli non sono permessi.

Indirizzo e-mail: \*

Un indirizzo e-mail valido. Il sistema invierà tutte le e-mail a questo indirizzo. L'indirizzo e-mail non sarà pubblico e verrà utilizzato soltanto se desideri ricevere una nuova password o se vuoi ricevere notizie e avvisi via e-mail.

Password: \*

Conferma la password: \*

Fornire una password per il nuovo account in entrambi i campi

Stato:

☐ Bloccato

☒ Attivo

Ruoli:

☒ Utente autenticato

☐ admin

☐ Notifica l'utente del nuovo profilo

Informazioni personali

Nome: \*

Nome vero

Cognome: \*

Cognome

Nazionalità: \*

Seleziona Paese

Impostazioni di linguaggio

Lingua:

☐ Inglese (English)

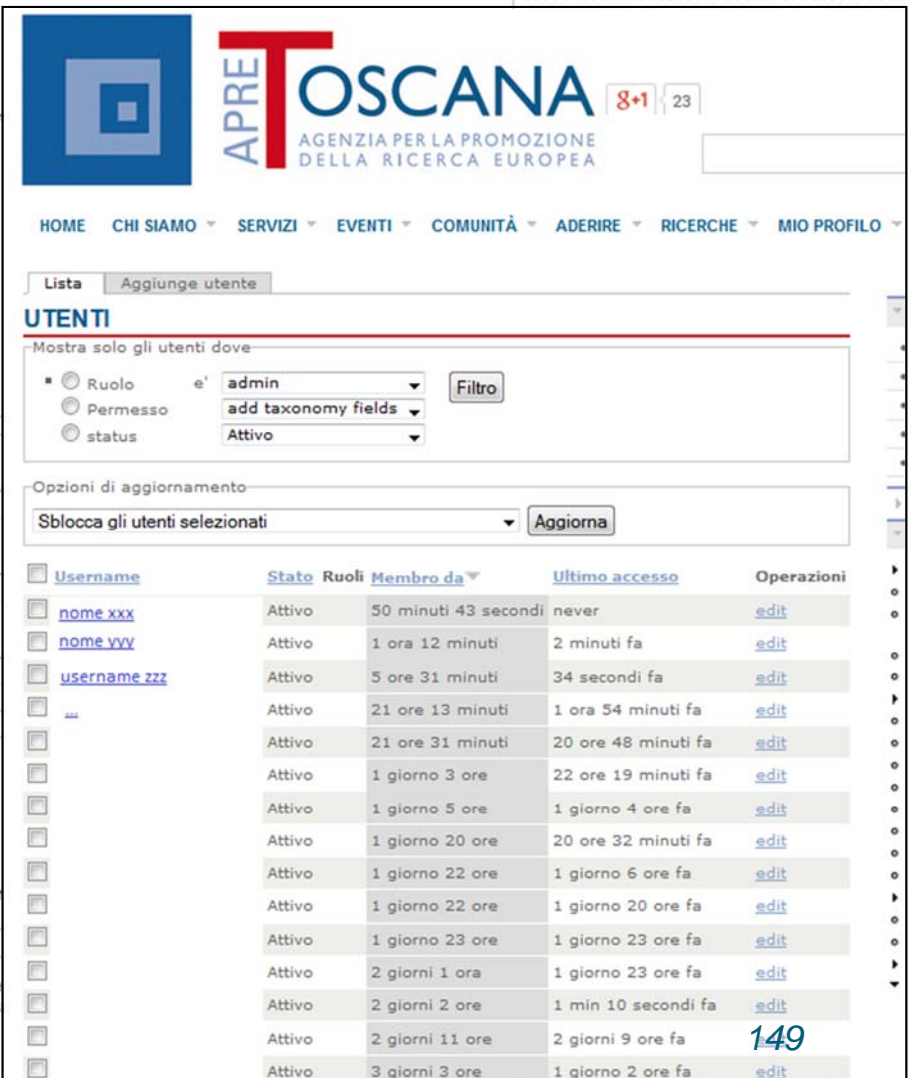
☒ Italiano (Italian)

Questo è il linguaggio che sarà usato per email e per l'interfaccia

Gruppi

You can subscribe one or more groups lists have access to private content, to the educational activity.

☐ Iscriviti General management.



APRE TOSCANA 8+1 23  
AGENZIA PER LA PROMOZIONE DELLA RICERCA EUROPEA

HOME CHI SIAMO SERVIZI EVENTI COMUNITÀ ADERIRE RICERCHE MIO PROFILO

Lista Aggiunge utente

### UTENTI

Mostra solo gli utenti dove

☒ Ruolo e' admin ☐ Permessi ☐ status

Opzioni di aggiornamento  
Sblocca gli utenti selezionati

Username	Stato	Ruoli	Membro da	Ultimo accesso	Operazioni
<input checked="" type="checkbox"/> <a href="#">nome xxx</a>	Attivo		50 minuti 43 secondi	never	<a href="#">edit</a>
<input checked="" type="checkbox"/> <a href="#">nome yyy</a>	Attivo		1 ora 12 minuti	2 minuti fa	<a href="#">edit</a>
<input checked="" type="checkbox"/> <a href="#">username zzz</a>	Attivo		5 ore 31 minuti	34 secondi fa	<a href="#">edit</a>
<input type="checkbox"/> <a href="#">...</a>	Attivo		21 ore 13 minuti	1 ora 54 minuti fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		21 ore 31 minuti	20 ore 48 minuti fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		1 giorno 3 ore	22 ore 19 minuti fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		1 giorno 5 ore	1 giorno 4 ore fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		1 giorno 20 ore	20 ore 32 minuti fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		1 giorno 22 ore	1 giorno 6 ore fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		1 giorno 22 ore	1 giorno 20 ore fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		1 giorno 23 ore	1 giorno 23 ore fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		2 giorni 1 ora	1 giorno 23 ore fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		2 giorni 2 ore	1 min 10 secondi fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		2 giorni 11 ore	2 giorni 9 ore fa	<a href="#">edit</a>
<input type="checkbox"/>	Attivo		3 giorni 3 ore	1 giorno 2 ore fa	<a href="#">edit</a>

149



# Drupal: amministrazione (7)

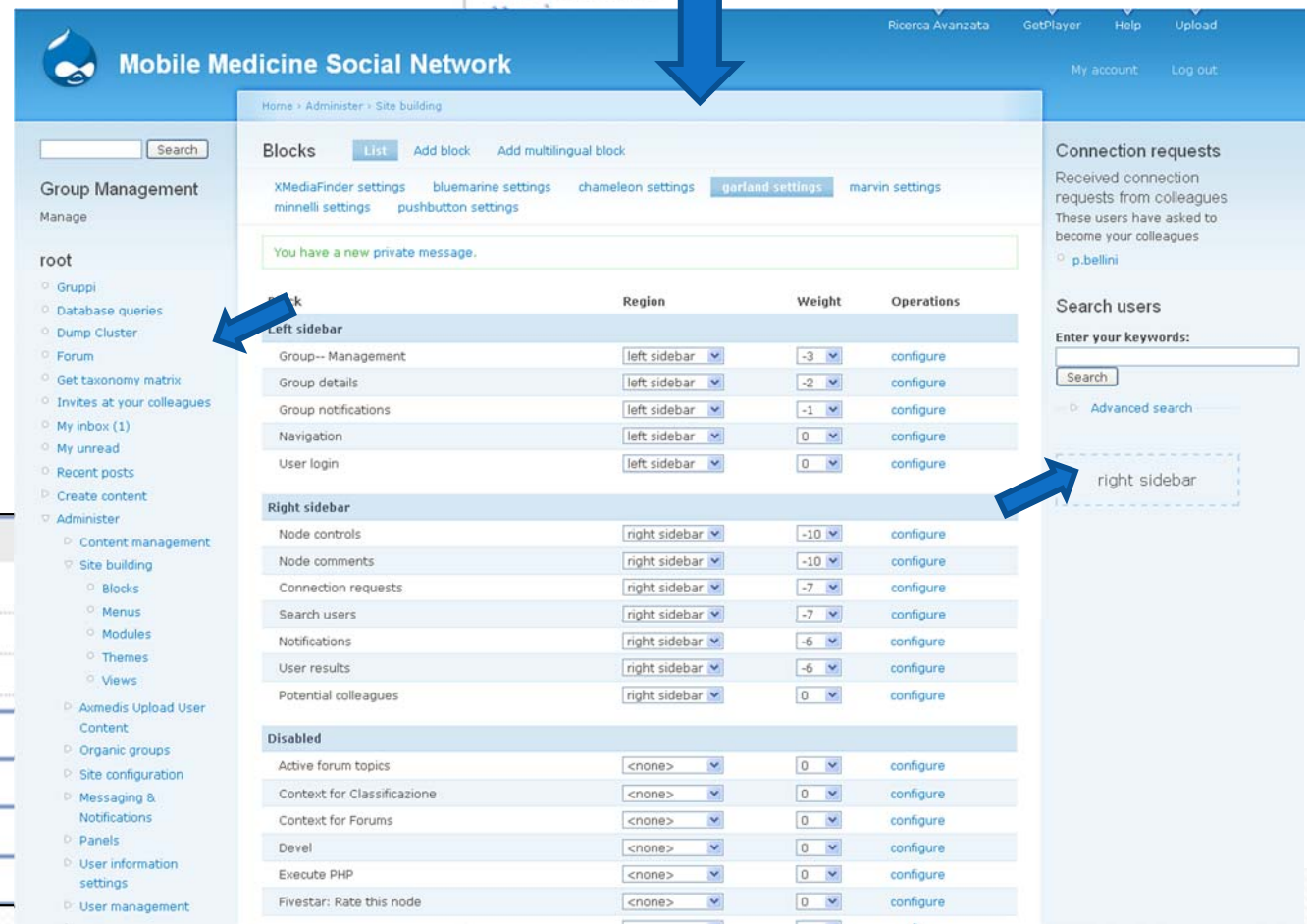
- Struttura del sito:
  - Blocchi: Una pagina generata da Drupal non è composta soltanto dal contenuto proposto, ma può essere integrata dai blocchi. (es: form di login, il menu di navigazione, etc.)

## Struttura del sito

Controlla l'aspetto del Vostro sito.

### Blocchi

Configura quale contenuto del blocco debba apparire nelle barre laterali del sito e in altre aree



**Mobile Medicine Social Network**

Home > Administer > Site building

**Blocks** List Add block Add multilingual block

XMediaFinder settings blumarine settings chameleon settings **garland settings** marvin settings minnelli settings pushbutton settings

You have a new private message.

Block	Region	Weight	Operations
<b>Left sidebar</b>			
Group-- Management	left sidebar	-3	configure
Group details	left sidebar	-2	configure
Group notifications	left sidebar	-1	configure
Navigation	left sidebar	0	configure
User login	left sidebar	0	configure
<b>Right sidebar</b>			
Node controls	right sidebar	-10	configure
Node comments	right sidebar	-10	configure
Connection requests	right sidebar	-7	configure
Search users	right sidebar	-7	configure
Notifications	right sidebar	-6	configure
User results	right sidebar	-5	configure
Potential colleagues	right sidebar	0	configure
<b>Disabled</b>			
Active forum topics	<none>	0	configure
Context for Classificazione	<none>	0	configure
Context for Forums	<none>	0	configure
Devel	<none>	0	configure
Execute PHP	<none>	0	configure
Fivestar: Rate this node	<none>	0	configure

**Connection requests**  
Received connection requests from colleagues  
These users have asked to become your colleagues  
p.bellini

**Search users**  
Enter your keywords:  
Search  
Advanced search



**CALENDARIO**

Marzo

L	M	M	G	V	S
3	4	5	6	7	8
10	11	12	13	14	15
17	18	19	20	21	22
24	25	26	27	28	29

**LOGIN**

Username: \*

Password: \*

Accedere

**CONTENUTI**

- Most Viewed (last 1000)
- Ultimi caricati
- Più votati

FROM TWITTER ...

NUVOLA DELLA KEYWORD

CLASSIFICAZIONE

GRUPPI



- ## Struttura del sito

Blocchi

[Menù](#)

Controlla il menu' di navigazione del sito, i link primari e secondari, cosi' come rinomina e riorganizza gli oggetti del menu'.

Salva

# Drupal: amministrazione (9)

- Struttura del sito:
  - **Moduli:**
    - L'elenco iniziale comprende i moduli del Drupal core. Ogni modulo contribuisce alla realizzazione del sito fornendo specifici servizi e funzionalità
    - Alcuni moduli hanno una sorta di propedeuticità, ovvero sono richiesti da altri moduli (es: il modulo "Comment" deve essere attivato perché il modulo "Forum" funzioni)

**Moduli**

▼ [Core - Informazioni opzionali](#)

Attivo	Nome	Versione	Descrizione
<input checked="" type="checkbox"/>	Forum	5.10	<p>Enables threaded discussions about general topics.</p> <p>Richiede: Taxonomy (<b>attivo</b>), Comment (<b>attivo</b>) Richiesto da: OG forum (<b>attivo</b>)</p>

**Struttura del sito**

Controlla l'aspetto del Vostro sito.

[Moduli](#)  
Attiva o disattiva moduli aggiuntivi (add-on) per il sito.

[Pagine](#)  
Add, edit and remove overridden system

Lista Disinstalla

**MODULI**


▼ [Core - Informazioni opzionali](#)

<input checked="" type="checkbox"/>	Forum	6.16	<p>Enables threaded discussions about general topics.</p> <p>Richiede: Taxonomy (<b>enabled</b>), Comment (<b>enabled</b>) Richiesto da: OG forum (<b>enabled</b>), luceneIndex (<b>enabled</b>)</p>
<input checked="" type="checkbox"/>	Help	6.16	<p>Manages the display of online help.</p>
<input checked="" type="checkbox"/>	Locale	6.16	<p>Adds language handling functionality and enables the translation of the user interface to languages other than English.</p> <p>Richiesto da: Date Locale (<b>disabled</b>), Internationalization (<b>enabled</b>), Internationalization tests (<b>disabled</b>), String translation (<b>enabled</b>), Language Switcher Dropdown (<b>enabled</b>), Language icons (<b>enabled</b>), Translation template extractor (<b>disabled</b>), Content translation (<b>enabled</b>), Block translation (<b>enabled</b>), CCK translation (<b>disabled</b>), Content type translation (<b>disabled</b>), Menu translation (<b>enabled</b>), Poll aggregate (<b>disabled</b>), Profile translation (<b>disabled</b>), Synchronize translations (<b>disabled</b>), Taxonomy translation (<b>disabled</b>), Views translation (<b>disabled</b>), Multilingual Taxonomy Export (<b>disabled</b>)</p>
<input checked="" type="checkbox"/>	Search	6.16	<p>Enables site-wide keyword searching.</p> <p>Richiesto da: Location Search (<b>disabled</b>), Project issue search index (<b>disabled</b>), Search Users (<b>enabled</b>)</p>
<input checked="" type="checkbox"/>	Taxonomy	6.16	<p>Enables the categorization of content.</p> <p>Richiesto da: Drupal Wiki taggin suggestions (<b>disabled</b>), Forum (<b>enabled</b>), GMap Taxonomy Markers (<b>disabled</b>), Taxonomy translation (<b>disabled</b>), Location Taxonomy (<b>disabled</b>), Multilingual Taxonomy Export (<b>disabled</b>), Taxonomy Notifications (<b>disabled</b>), Project releases (<b>disabled</b>), Project Solr (<b>disabled</b>), Tagadelic (<b>enabled</b>), Tagging (<b>disabled</b>), Tagging opencalais suggestions (<b>disabled</b>), Tagging full-vocabular suggestion (<b>disabled</b>), Taxonomy context (<b>enabled</b>), Taxonomy Manager (<b>enabled</b>), Taxonomy Super Select (<b>enabled</b>), Taxonomy Import/Export via XML (<b>enabled</b>), Community Tags (<b>enabled</b>), Views translation (<b>disabled</b>), OG forum (<b>enabled</b>), Project package (<b>disabled</b>), Project usage (<b>disabled</b>), Semantic Tags (<b>enabled</b>), Tagadelic Views (<b>disabled</b>), Tagging AlchemyAPI suggestions (<b>disabled</b>), Tagging vocab extractor suggestions (<b>disabled</b>), luceneIndex (<b>enabled</b>)</p>
<input checked="" type="checkbox"/>	Upload	6.16	<p>Allows users to upload and attach files to content.</p> <p>Richiesto da: Comment upload (<b>enabled</b>), Project issue tracking (<b>enabled</b>), Project releases (<b>disabled</b>), Project issue generator (<b>disabled</b>), Project issue search index (<b>disabled</b>), Project package (<b>disabled</b>), Project usage (<b>disabled</b>)</p>



# Drupal: concetto di modulo

- Per comprendere le funzionalità fornite da ogni modulo è sempre consigliabile fare riferimento al sito Drupal, in cui si trovano opportuni commenti, descrizioni ed esempi di utilizzo (sia per i moduli che fanno parte del Drupal core che per quelli aggiuntivi):
  - <http://drupal.org/project/Modules/>


**Drupal**
[Documentation](#)
[Download](#)

Home » Download » Modules

**Maintainers for OAI-PMH Module**

- [rjerome](#) - 56 commits  
last: 29 weeks ago, first: 3 years ago

[View all committers](#)


**Issues for OAI-PMH Module**

To avoid duplicates, please search before submitting a new issue.

  
 [Advanced search](#)
  
**All issues**  
 14 open, 16 total  
**Bug reports**  
 4 open, 5 total  
[Subscribe via e-mail](#)  
[Issue statistics](#)  
 Oldest open issue: 5 Jan 07

**Recent issues**

- OAI Module for CCK
- OAI-PMH Module for CCK Fields
- harvesting for any node type ...


**UNIVERSITÀ DEGLI STUDI FIRENZE**  
 Dipartimento di Ingegneria dell'Informazione

**OAI-PMH Module**

[View](#)
[CVS instructions](#)

  
[rjerome](#) - August 5, 2006 - 01:14
 

Modules · Syndication

This module provides an **Open Archives Initiative Protocol for Metadata Harvesting** (OAI-PMH), interface to the **Bibliography Module**.

Currently it only exposes metadata from the biblio module; however if there was demand I believe that it could be made configurable to expose other node types as well.

*"This module was inspired by and I have used some code from <http://physnet.uni-oldenburg.de/oai/> written by Heinrich Stamerjohanns."*

  
**Downloads**
  
**Recommended releases**

Version	Downloads	Date	Links
<a href="#">6.x-1.2</a>	<a href="#">Download (14.15 KB)</a>	2009-Jun-22	<a href="#">Notes</a>
<a href="#">5.x-1.3</a>	<a href="#">Download (14.22 KB)</a>	2008-Jun-24	<a href="#">Notes</a>
<a href="#">4.7.x-1.1</a>	<a href="#">Download (13.78 KB)</a>	2007-Jun-21	<a href="#">Notes</a>

  
**Development releases**

Version	Downloads	Date	Links
<a href="#">6.x-1.x-dev</a>	<a href="#">Download (14.15 KB)</a>	2009-Jun-23	<a href="#">Notes</a>



# Drupal: amministrazione (10)

- Struttura del sito:
  - **Moduli:**
    - L'elenco iniziale comprende i moduli del Drupal core. Ogni modulo contribuisce alla realizzazione del sito fornendo specifici servizi e funzionalità
    - Alcuni moduli hanno una sorta di propedeuticità, ovvero sono richiesti da altri moduli (es: il modulo "Comment" deve essere attivato perché il modulo "Forum" funzioni)

**Moduli**

▼ Core - Informazioni opzionali

Attivo	Nome	Versione	Descrizione
<input checked="" type="checkbox"/>	Forum	5.10	<p>Enables threaded discussions about general topics.</p> <p>Richiede: Taxonomy (attivo), Comment (attivo) Richiesto da: OG forum (attivo)</p>

**Struttura del sito**

Controlla l'aspetto del Vostro sito.

[Moduli](#)  
Attiva o disattiva moduli aggiuntivi (add-on) per il sito.

[Pagine](#)  
Add, edit and remove overridden system

Lista Disinstalla

**MODULI**

▼ Core - Informazioni opzionali

<input checked="" type="checkbox"/>	Forum	6.16	<p>Enables threaded discussions about general topics.</p> <p>Richiede: Taxonomy (enabled), Comment (enabled) Richiesto da: OG forum (enabled), luceneIndex (enabled)</p>
<input checked="" type="checkbox"/>	Help	6.16	<p>Manages the display of online help.</p>
<input checked="" type="checkbox"/>	Locale	6.16	<p>Adds language handling functionality and enables the translation of the user interface to languages other than English.</p> <p>Richiesto da: Date Locale (disabled), Internationalization (enabled), Internationalization tests (disabled), String translation (enabled), Language Switcher Dropdown (enabled), Language icons (enabled), Translation template extractor (disabled), Content translation (enabled), Block translation (enabled), CCK translation (disabled), Content type translation (disabled), Menu translation (enabled), Poll aggregate (disabled), Profile translation (disabled), Synchronize translations (disabled), Taxonomy translation (disabled), Views translation (disabled), Multilingual Taxonomy Export (disabled)</p>
<input checked="" type="checkbox"/>	Search	6.16	<p>Enables site-wide keyword searching.</p> <p>Richiesto da: Location Search (disabled), Project issue search index (disabled), Search Users (enabled)</p>
<input checked="" type="checkbox"/>	Taxonomy	6.16	<p>Enables the categorization of content.</p> <p>Richiesto da: Drupal Wiki taggin suggestions (disabled), Forum (enabled), GMap Taxonomy Markers (disabled), Taxonomy translation (disabled), Location Taxonomy (disabled), Multilingual Taxonomy Export (disabled), Taxonomy Notifications (disabled), Project releases (disabled), Project Solr (disabled), Tagadelic (enabled), Tagging (disabled), Tagging opencalais suggestions (disabled), Tagging full-vocabular suggestion (disabled), Taxonomy context (enabled), Taxonomy Manager (enabled), Taxonomy Super Select (enabled), Taxonomy Import/Export via XML (enabled), Community Tags (enabled), Views translation (disabled), OG forum (enabled), Project package (disabled), Project usage (disabled), Semantic Tags (enabled), Tagadelic Views (disabled), Tagging AlchemyAPI suggestions (disabled), Tagging vocab extractor suggestions (disabled), luceneIndex (enabled)</p>
<input checked="" type="checkbox"/>	Upload	6.16	<p>Allows users to upload and attach files to content.</p> <p>Richiesto da: Comment upload (enabled), Project issue tracking (enabled), Project releases (disabled), Project issue generator (disabled), Project issue search index (disabled), Project package (disabled), Project usage (disabled)</p>

# Drupal: Vocabolari e termini

- Drupal fornisce un ottimo meccanismo per catalogare il contenuto, ovvero la **tassonomia**, attraverso il modulo **Taxonomy**. Esso viene attivato di default, per cui fin dall'installazione possiamo accedere alla **gestione della tassonomia** (Percorso: Amministra > Gestione dei contenuti > Taxonomy)
- Alla base del concetto di tassonomia di Drupal, ci sono due definizioni:
  - **Vocabolario**: un concetto che viene descritto attraverso i termini
  - **Termine**

Ad esempio, il vocabolario "Sport" ha come possibili termini calcio, nuoto, pallavolo
- Per ogni tipologia di contenuto offerta attraverso il sito è possibile assegnare più vocabolari, in questo modo la gestione della tassonomia di Drupal permette di creare categorie, sottosezioni, gerarchie, categorie incrociate, sistemi di tagging, etc.



# Drupal: Temi

- I temi Drupal servono per la presentazione del sito (riguardano gli aspetti grafici)
- Drupal offre alcuni template aggiuntivi, oltre a quello di default, che permettono di personalizzare l'interfaccia grafica offerta ai vostri visitatori (Percorso: Amministra > Struttura del sito > Temi)
- Esistono temi Drupal aggiuntivi rispetto a quelli di default (basta scaricarli dal sito ufficiale ed inserirli nella cartella **/sites/all/themes/**)
- E' possibile creare un tema personale



# Drupal: come è fatto un modulo

- In ogni modulo sono fondamentalmente presenti almeno i seguenti file:
  - NomeModulo.info: contiene la descrizione
  - NomeModulo.install: serve per la definizione delle tabelle nel database (creazione di nuove tabelle nel DB, modifica di vecchie, etc.)
  - NomeModulo.module: è un file php (salvato poi come .module), contiene le funzioni (hook) che verranno chiamate
  - Readme.txt: contiene eventuali note aggiuntive

# Esempio modulo OAI-PMH (oai2.info)

- ; \$Id: oai2.info,v 1.1.2.2.2.1 2008/12/15 22:29:04 rjerome Exp \$  
name = OAI2  
description = This module provides Open Archives 2 protocol  
access to the information stored by the Biblio module  
dependencies[] = biblio  
core = 6.x  
package = Biblio

; Information added by drupal.org packaging script on 2009-06-22  
version = "6.x-1.2"  
core = "6.x"  
project = "oai2"  
timestamp = "1245678440"

# Esempio modulo OAI-PMH (oai2.module)

```
<?php// $Id: oai2.module,v 1.9.2.4.2.5 2009/06/22 13:21:34 rjerome
Exp $
/** *   oai2.module for Drupal
* *   Copyright (C) 2006  Ron Jerome
* *   This program is free software; [...]
*/
function oai2_help($section) {
  switch ($section) {
    case 'admin/modules#description':
      return t('This module provides Open Archives 2 protocol access to
the information stored by the Biblio module' );
    }
  }
function oai2_node_name($node) {
  return t('oai2');
} [...]
```



# Esempio modulo OAI-PMH (oai2.install)

```
<?PHP
// $Id: oai2.install,v 1.4.2.1.2.3 2008/12/16 19:11:38 rjerome Exp $
function oai2_schema(){
  $schema['oai2_metadata_formats'] = array(
    'fields' => array(
      'name' => array('type' => 'varchar','length' => 100,'not null' => true,'default' => ''),
      'metadata_prefix' => array('type' => 'varchar','length' => 100,'not null' => true,'default' => ''),
      'schema' => array('type' => 'varchar','length' => 255,'not null' => true,'default' => ''),
      'metadata_namespace' => array('type' => 'varchar','length' => 255,'not null' => true,'default' => ''),
      'record_prefix' => array('type' => 'varchar','length' => 100,'not null' => true,'default' => ''),
      'record_namespace' => array('type' => 'varchar','length' => 255,'not null' => true,'default' => ''),
    ),
    'primary key' => array('name')
  );
}
```

## • [...]

```
function oai2_update_1() {
  $result = array();
  $result[] = update_sql("ALTER TABLE {oai2_tokens} MODIFY id VARCHAR(13) NOT NULL DEFAULT 0");
  return $result;
}

function oai2_update_6000() {
  $result = array();
  $result[] = update_sql("ALTER TABLE {oai2_tokens} MODIFY id VARCHAR(32) NOT NULL DEFAULT 0");
  return $result;
}
```

# Esempio: og.info

name = Organic groups

description = "Enable users to create and manage groups. OG Views integration module is recommended for best experience."

package = "Organic groups"

; Information added by drupal.org packaging script on 2010-03-16

version = "6.x-2.1"

core = 6.x

project = "og"

timestamp = "1268751911"

MODULI		
Active	Name	Version Description
<input checked="" type="checkbox"/>	OG forum	6.x-2.2 Creates forums within organic groups. Depends on: Forum ( <b>enabled</b> ), Organic groups ( <b>enabled</b> ), Taxonomy ( <b>enabled</b> ), Comment ( <b>enabled</b> ) Required by: luceneIndex ( <b>enabled</b> )
<input checked="" type="checkbox"/>	Organic groups	6.x-2.1 Enable users to create and manage groups. OG Views integration module is recommended for best experience. Required by: Organic groups access control ( <b>enabled</b> ), Organic groups actions ( <b>disabled</b> ), OG forum ( <b>enabled</b> ), Organic groups HomePage Settings ( <b>disabled</b> ), Organic Groups Notifications ( <b>enabled</b> ), Organic groups panels. ( <b>enabled</b> ), Organic groups user roles ( <b>enabled</b> ), Organic groups Views integration ( <b>enabled</b> ), luceneIndex ( <b>enabled</b> )
<input checked="" type="checkbox"/>	Organic groups access control	6.x-2.1 Enable access control for private posts and private groups. Depends on: Organic groups ( <b>enabled</b> )
<input type="checkbox"/>	Organic groups actions	6.x-2.1 Provides actions for use with Organic Groups and Trigger. Depends on: Organic groups ( <b>enabled</b> )
<input checked="" type="checkbox"/>	Organic Groups Notifications	6.x-2.1 Allows users to receive notifications and subscribe to group content. Depends on: Organic groups ( <b>enabled</b> ), Notifications ( <b>enabled</b> ), Content Notifications ( <b>enabled</b> ), Notifications Lite ( <b>enabled</b> ), Messaging ( <b>enabled</b> ), Token ( <b>enabled</b> )
<input checked="" type="checkbox"/>	Organic groups panels.	6.x-2.0 Enable group admins to create panel pages. Depends on: Organic groups ( <b>enabled</b> ), Panels ( <b>enabled</b> ), Page manager ( <b>enabled</b> ), Chaos tools ( <b>enabled</b> )
<input checked="" type="checkbox"/>	Organic groups user roles	6.x-4.1 Assign additional user roles to members of organic groups within the group context. Depends on: Organic groups ( <b>enabled</b> ), Organic groups Views integration ( <b>enabled</b> ), Views ( <b>enabled</b> )
<input checked="" type="checkbox"/>	Organic groups Views integration	6.x-2.1 <b>Highly recommended..</b> Use Views to search and display organic groups. Depends on: Organic groups ( <b>enabled</b> ), Views ( <b>enabled</b> ) Required by: Organic groups user roles ( <b>enabled</b> )



# Esempio: og.install

```

/**
 * Definition of hook_schema();
 */
function og_schema() {
  $schema = array();

  $schema['og'] = array(
    'description' => 'Stores information about each group.',
    'fields' => array(
      'nid' => array(
        'description' => "The group's [node].nid.",
        'type' => 'int',
        'size' => 'normal',
        'not null' => TRUE,
      ),
      'og_selective' => array(
        'description' => 'Determines how subscribers...',
        'type' => 'int',
        'size' => 'normal',
        'not null' => TRUE,
        'default' => 0,
      ),
      'og_description' => array(
        'description' => 'Group description. Shows...',
        'type' => 'varchar',
        'length' => 255,
        'not null' => FALSE,
      ),
      'og_theme' => array(
        'description' => 'The group specific theme',
        'type' => 'varchar',
        'length' => 255,
        'not null' => FALSE,
      ),
      ...
      'og_main' => array(
        'description' => 'Main page.',
        'type' => 'int',
        'size' => 'normal',
        'not null' => TRUE,
        'default' => 0,
      ),
    ),
  );
  ...
  'primary key' => array('nid'),
};

```

MySQL Query Browser - Connection: root@localhost:3306 / drupal

Query: `SELECT * FROM og o;`

Resultset 5

nid	og_selective	og_description	og_theme	og_register	og_directory	og_language	og_private	og_blockweight	og_main
2859	1	Coordinator: ...		1	1	it	0	0	0
3480	1	Coordinator: na	HULL	1	1		0	0	0
2862	1	Coordinator: ...		1	1	it	0	0	0
3481	1	Coordinator: na	HULL	1	1		0	0	0
3479	1	Coordinator: na	HULL	1	1		0	0	1
2864	1	Coordinator: ...		1	1	it	0	-10	0
3478	1	Coordinator: na	HULL	1	1		0	0	0
2867	1	Coordinator: na		1	1	it	0	0	0
3474	1	coordinator: na	HULL	1	1		0	0	0
3160	1	To organize a...		1	1	it	0	10	3
3491	1		HULL	1	1		0	0	0
3161	1	To discuss a...		1	1	it	0	10	4
3232	1	Coordinator: J...		1	1	it	0	-9	0
3489	1	Coordinator: na	HULL	1	1		0	0	0
3261	1	Coordinator: ...		1	1	it	0	-8	0
3275	1	Coordinator: na		1	1	it	0	0	1
3427	2	Description	HULL	0	0	it	0	-10	0
3344	1	Coordinator: ...		1	1	it	0	0	0
3362	0	Co-ordinators...		1	1	it	0	0	0
3364	1	Coordinator: ...		1	1	it	0	0	0
3476	1	Coordinator: na	HULL	1	1		0	0	0
3477	1	Coordinator: na	HULL	1	1		0	0	0

26 rows fetched in 0.0195s (0.0005s)

Schemata: og  
 nid  
 og\_selective  
 og\_description  
 og\_theme  
 og\_register  
 og\_directory  
 og\_language  
 og\_private  
 og\_blockweight  
 og\_main  
 og\_access\_post

Syntax: Functions Params Trx  
 Data Definition Statements  
 Data Manipulation Statements  
 MySQL Utility Statements  
 MySQL Transactional and Locking ...  
 Database Administration Statements  
 Replication Statements  
 SQL Syntax for Prepared Statements



# Esempio: og.install - og\_uninstall()

```
function og_uninstall() {  
  drupal_uninstall_schema('og');
```

```
// In case og_notifications has never been enabled, the vestigial  
// og_uid_global table will still be around.  
if (variable_get('og_notifications_update_required', FALSE)) {  
  db_query('DROP TABLE {og_uid_global}');
```

```
// Delete variables
```

```
$variables = array(  
  'og_help', 'og_block_cnt_2', 'og_block_cnt_3',  
  'og_audience_checkboxes', 'og_omitted', 'og_content_type',  
  'og_audience_required',  
  'og_visibility_directory', 'og_visibility_registration',  
  'og_home_page_view', 'og_email_max', 'og_node_types',  
  'og_admin_email_body', 'og_email_notification_pattern',  
  'og_approve_user_body', 'og_approve_user_subject',  
  'og_deny_user_body', 'og_deny_user_subject',  
  'og_invite_user_body', 'og_invite_user_subject',  
  'og_new_admin_body', 'og_new_admin_subject',  
  'og_new_node_body', 'og_new_node_subject',  
  'og_request_user_body', 'og_request_user_subject',  
  'og_notifications_update_required', 'og_notifications_co  
);
```

```
foreach (node_get_types() as $type_name => $type) {  
  $variables[] = 'og_max_groups_' . $type_name;  
}
```

```
foreach ($variables as $variable) {  
  variable_del($variable);  
}
```

Disinstalla

**MODULI**

Disinstalla	Nome	Descrizione
<input type="checkbox"/>	Administration menu	Provides a dropdown menu to most administrative tasks and other common destinations (to users with the proper permissions).
<input type="checkbox"/>	Annotations	Allows users annotate a media
<input type="checkbox"/>	Axrelations Graph	Create a relations graph of a node with the other elements in the database.
<input type="checkbox"/>	Eclap Mailman	
<input type="checkbox"/>	Fivestar Comments	Rate nodes by leaving comments.
<input type="checkbox"/>	Mailman Manager	Allows members to subscribe and change their subscriptions to Mailman mailing lists.
<input type="checkbox"/>	Playlists	Alows users to compose playlists
<input checked="" type="checkbox"/>	Organic groups	Enable users to create and manage groups. OG Views integration module is recommended for best experience.
<input type="checkbox"/>	Potential Friends	Mostra i potenziali colleghi di ogni utente
<input type="checkbox"/>	Promoted	Alows users to mark some contents as their promoted
<input type="checkbox"/>	Register Axuid	Modulo che associa l'Uid Drupal all'Axuid Axmedis

Disinstalla


# Drupal: concetto di hook (1)

- Gli hook possono essere visti come ‘eventi interni a Drupal’, sono chiamati anche *callbacks*
- Sono definiti all’interno del codice php di ogni modulo
- Hanno una particolare nomenclatura (NomeModulo\_nomehook)
- Esempio: quando un utente si logga, drupal automaticamente esegue tutte le funzionalità legate all’hook ‘user’:
  - comment\_user() in the comment module,
  - locale\_user() in the locale module,
  - node\_user() in the node module,and any other similarly named functions will be called



# Drupal: concetto di hook (2)

- Se si crea un nuovo modulo ('nuovo') e si definisce un hook 'user', quando utente si logga Drupal richiama automaticamente anche il seguente hook:
  - nuovo\_user
- Gli hook definiti nei moduli Drupal sono tutti descritti ampiamente nel sito di riferimento (si faccia attenzione alla versione di Drupal in uso):
  - Drupal api:  
<http://api.drupal.org/api>



The screenshot shows the Drupal API reference page. At the top, there's a blue header with the Drupal logo and the text 'Drupal API'. To the right of the header are links: 'Report a bug', 'Developing for Drupal', and 'Drupal.org'. Below the header, there's a search box labeled 'Search Drupal 6' with a dropdown menu for 'Function, file, or topic:' and a 'Search' button. To the left of the main content area, there's a sidebar with a list of links: 'Drupal 6', 'Constants', 'Files', 'Functions', 'Globals', and 'Topics'. Below this list is a section titled 'Relevant references' with links to 'PHP', 'MySQL, Postgres', 'HTML', 'CSS', 'JavaScript', and 'jQuery'. The main content area is titled 'API reference' and has tabs for 'Drupal 4.6', 'Drupal 4.7', 'Drupal 5', 'Drupal 6', and 'Drupal 7'. The 'Drupal 6' tab is selected. The main content area contains a welcome message and a list of components of Drupal, including 'Module system (Drupal hooks)', 'Database abstraction layer', 'Menu system', 'Form generation', 'File upload system', 'Search system', 'Node access system', 'Theme system', 'Constants', and 'Global variables'. There is also a section for 'Example modules' with links to 'How to define blocks' and 'How to define a content filter'.



# Drupal: I principali hook

- `NomeModulo_help()`: permette a Drupal di avere informazioni riguardo al modulo che si sta scrivendo
- `NomeModulo_perm()`: definisce i permessi disponibili per il modulo
- `NomeModulo_menu()`: definisce le pagine associate al modulo
- `NomeModulo_block()`: definisce i blocchi associati al modulo
- `NomeModulo_settings()`: permette di configurare il modulo affinché il suo comportamento risponda alle proprie esigenze

# Esempio: og.module (1)

- hook\_perm()

```
/**
 * Implementation of hook_perm().
 */
function og_perm() {
  // return array('administer organic groups');
  return array('administer organic groups', 'send new group request', 'view group content', 'og_broadcast_access');
}
```

- In Administer > Permissions:

Permission	anonymous user	authenticated user	admin
<b>og module</b>			
administer organic groups	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
og_broadcast_access	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
send new group request	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
view group content	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

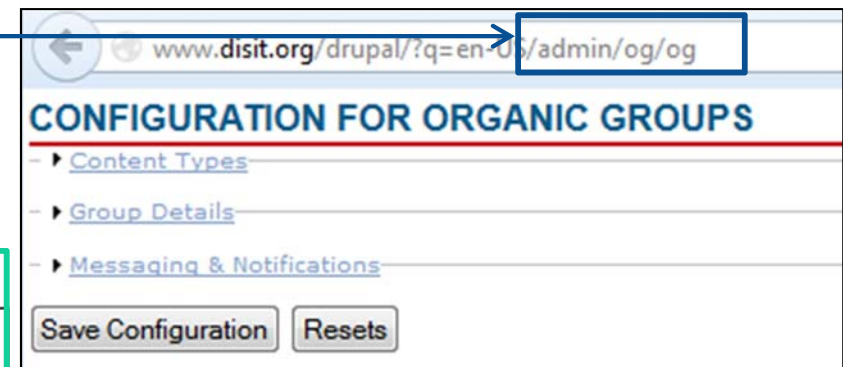
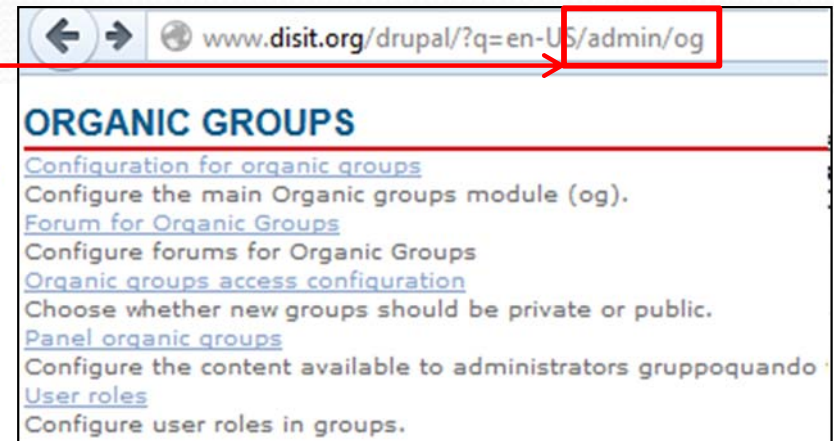
# Esempio: og.module (2)

- hook\_menu()

```
$items['admin/og'] = array(
  'title' => 'Organic groups',
  'description' => 'Administer the suite of Organic groups modules.',
  'position' => 'right',
  'weight' => -5,
  'page callback' => 'system_admin_menu_block_page',
  'access arguments' => array('administer site configuration'),
  'file' => 'system.admin.inc',
  'file path' => drupal_get_path('module', 'system'),
);
$items['admin/og/og'] = array(
  'page callback' => 'drupal_get_form',
  'page arguments' => array('og_admin_settings'),
  'title' => 'Organic groups configuration',
  'access arguments' => array('administer site configuration'),
  'description' => 'Configure the main Organic groups module (og).',
  'file' => 'og.admin.inc',
  'file path' => drupal_get_path('module', 'og'). '/includes',
  'weight' => -5,
);
```

## File system.admin.inc

```
function system_admin_menu_block_page() {
  $item = menu_get_item();
  if ($content = system_admin_menu_block($item)) {
    $output = theme('admin_block_content', $content);
  }
  else {
    $output = t('You do not have any administrative items.');
```





# Esempio: menu\_item\_types

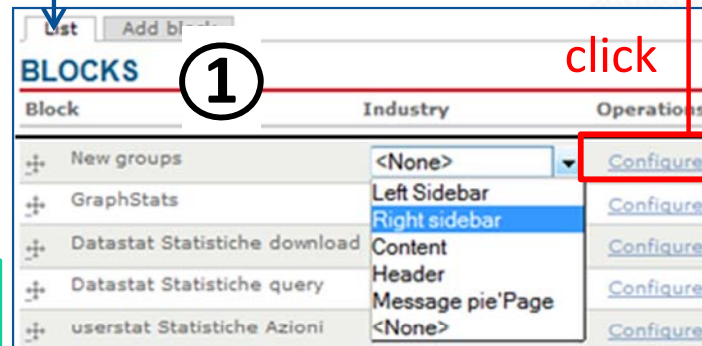
- [https://api.drupal.org/api/drupal/includes!menu.inc/group/menu\\_item\\_types/6](https://api.drupal.org/api/drupal/includes!menu.inc/group/menu_item_types/6)

Name ▲	Location	Description
MENU_CALLBACK	includes/ menu.inc	Callbacks simply register a path so that the correct function is fired when the URL is accessed. They are not shown in the menu.
MENU_DEFAULT_LOCAL_TASK	includes/ menu.inc	Every set of local tasks should provide one "default" task, that links to the same path as its parent when clicked.
MENU_LOCAL_TASK	includes/ menu.inc	Local tasks are rendered as tabs by default. Use this for menu items that describe actions to be performed on their parent item. An example is the path "node/52/edit", which performs the "edit" task on "node/52".
MENU_NORMAL_ITEM	includes/ menu.inc	Normal menu items show up in the menu tree and can be moved/hidden by the administrator. Use this for most menu items. It is the default value if no menu item type is specified.
MENU_SUGGESTED_ITEM	includes/ menu.inc	Modules may "suggest" menu items that the administrator may enable. They act just as callbacks do until enabled, at which time they act like normal items. Note for the value: 0x0010 was a flag which is no longer used, but this way the values...

# Esempio: og.module (3)

/\*\*  
 \* Implementation of hook\_block().  
 \*/  
function og\_block(\$op = 'list', \$delta = 0, \$edit = array()) {  
 if (\$op == 'list') {  
 // [...]  
 \$blocks[3]['info'] = t('New groups');  
 \$blocks[3]['cache'] = BLOCK\_CACHE\_PER\_USER;  
 return \$blocks;  
 }  
 elseif (\$op == 'view') {  
 switch (\$delta) {  
 // [...]  
 case 3:  
 return og\_block\_new();  
 }  
 }  
 elseif (\$op == 'configure') {  
 switch (\$delta) {  
 // [...]  
 case 3:  
 return array('og\_block\_cnt' => array('#type' => 'textfield',  
 '#title' => t('Maximum number of groups to show'),  
 '#default\_value' => variable\_get("og\_block\_cnt\_\$delta", 10),  
 '#size' => 5,  
 '#maxlength' => 255));  
 }  
 }  
 elseif (\$op == 'save') {  
 switch (\$delta) {  
 // [...]  
 case 3:  
 if (isset(\$edit['og\_block\_cnt'])) {  
 variable\_set("og\_block\_cnt\_\$delta", \$edit['og\_block\_cnt']);  
 }  
 break;  
 }  
 }  
}

• og\_panels\_form



## Visualizzazione del blocco



## BLOCK 'NEW GROUPS'

### ▼ Collapsible

Block collapse behavior:

- ☐ None.
- ☒ Collapsible, expanded by default.
- ☐ Collapsible, collapsed by default.
- ☐ Collapsible, collapsed all the time.

### ► Multilingual settings

### ▼ Specific settings block

Title Block:

New Title

Override the default title for the block. Use <none> to display no title, or leave blank to use the default block title.

Maximum number of groups to show:

25

### ▼ Visibility user specific settings

Custom visibility settings:

- ☒ Users can not check to see whether or not this block.
- ☐ Show this block by default, but allows individual users to hide it.
- ☐ Hide this block by default, but allows individual users show it.

Allows individual users to customize the visibility of this block in their account settings.

### ▼ Role specific visibility settings

Show block for specific roles:

- ☒ admin
- ☐ anonymous user
- ☒ authenticated user

Show this block only selected roles. If you have not selected any role, the block will be visible to all users.

### ▼ Specific settings for page visibility

Show block on specific pages:

- ☒ Show on every page except the listed pages.
- ☐ Show on only the listed pages.
- ☐ Show if the following PHP code returns TRUE (PHP mode, experts only).

Pages:

Enter one page per line as Drupal paths. The character '\*' is a joker. Rooting for example are blog page and blog blog-wildcard for every blog. <front> the first page (front page). If you choose the mode PHP, you can put PHP code between <?php ?>. Note that PHP does properly can damage your Drupal site.

CKEditor: the ID for excluding or including this element is admin/build/block/configure/og/:pages.

Save block



# Esempio: og\_panels.module (1)

## • og\_panels\_form

```
function og_panels_form(&$form_state, $group_node, $did = NULL) {  
  //[...]  
  $form['page_title'] = array(  
    '#title' => t('Page title'),  
    '#type' => 'textfield',  
    '#required' => $display->default_page ? FALSE : TRUE,  
    '#default_value' => $display->page_title,  
    '#description' => t('This is the title of the page and of the tab.'),  
    '#size' => 32,  
  );  
  $form[$display->default_page ? 'path_placeholder' : 'path'] = array(  
    '#title' => t('Path'),  
    '#type' => 'textfield',  
    '#default_value' => $display->default_page ? '' : $display->path,  
    '#field_prefix' => url('node/'. $group_node->nid, array('absolute' => TRUE)) . '/',  
    '#description' => $display->default_page ? t('This page is currently your default  
    group home page and has no configurable path.') :  
    t('An optional path alias for this page.'),  
    '#disabled' => $display->default_page,  
    '#size' => 32,  
    '#access' => module_exists('path'),  
  );  
  //[...]  
  $form['show_blocks'] = array(  
    '#title' => t('Show blocks'),  
    '#type' => 'checkbox',  
    '#default_value' => isset($display->show_blocks) ? $display->show_blocks : TRUE,  
    '#description' => t('If unchecked, the standard group blocks will not be  
    shown unless you place them into your page content.  
    This gives admin more control over page presentation.'),  
  );  
  $form['published'] = array(  
    '#type' => 'checkbox',  
    '#title' => t('Published'),  
    '#default_value' => ($display->did) ? $display->published : TRUE,  
    '#description' => t('If unchecked, this page is only accessible  
    by group or site administrators.  
    That's useful while you are configuring the page.'),  
  );  
  $form['submit'] = array(  
    '#type' => 'submit',  
    '#value' => $did ? t('Update page') : t('Create page'),  
  );  
  $form['did'] = array('#type' => 'value', '#value' => $did);  
  $form['nid'] = array('#type' => 'value', '#value' => $group_node->nid);  
  $form['tab_num'] = array('#type' => 'value', '#value' => $display->tab_num);  
  return $form;  
}
```

## • og\_panels\_menu()

```
$items['node/%node/og_panels/form'] = array(  
  'page callback' => 'drupal_get_form',  
  'page arguments' => array('og_panels_form', 1),  
  'title callback' => 'og_panels_tab_edit_title',  
  'title arguments' => array(4),  
  'access callback' => 'og_panels_admin_root_content',  
  'access arguments' => array(1),  
  'type' => MENU_LOCAL_TASK,  
  'weight' => 1,  
);
```

3

Wall Edit Who online Members Blog Log Pages Objects Forum  
Broadcast Group Designer Devel  
Web Pages List Add Web Page Panels List Add new panel  
**GENERAL MANAGEMENT**  
Page title: \*  
This is the title of the page and of the tab.  
☒ Show blocks  
If unchecked, the standard group blocks will not be shown unless you place them into your page content. This gives admin more control over page presentation.  
☒ Published  
If unchecked, this page is only accessible by group or site administrators. That's useful while you are configuring the page.  
Create page

- ④ • og\_panels\_form\_validate(){...}
- ⑤ • og\_panels\_form\_submit(){...}
- ⑥ • Creazione effettiva del pannello



# Esempio: og\_panels.module (2)

## ④ • og\_panels\_form\_validate

```
function og_panels_form_validate($form, &$amp;form_state) {
  $pathblacklist = array('view', 'edit', 'delete', 'outline', 'load', 'render', 'clone');
  if (in_array($form_state['values']['path'], $pathblacklist)) {
    form_error($form['path'], t('%path is a reserved system path, and cannot be
    used for a group page. Please enter another path.',
    array('%path' => $form_state['values']['path'])));
  }
  else if (preg_match("/[^\A-Za-z0-9-]/", $form_state['values']['path'])) {
    form_error($form['path'], t('Panel paths may only contain alphanumeric
    characters and dashes.'));
  }
  else if (db_result(db_query("SELECT path FROM {og_panels} WHERE path = '%s'
    AND did <> %d AND nid = %d", $form_state['values']['path'],
    $form_state['values']['did'], $form_state['values']['nid']))) {
    form_error($form['path'], t("That path is currently in use by another
    one of your group's pages. Please enter another path."));
  }
}
```

## ⑤ • og\_panels\_form\_submit

```
function og_panels_form_submit($form, &$amp;form_state) {
  if ($form_state['values']['did']) {
    $sql = "UPDATE {og_panels} SET page_title = '%s', path = '%s', published = %d,
    show_blocks = %d WHERE did = %d";
    db_query($sql, $form_state['values']['page_title'], $form_state['values']['path'],
    $form_state['values']['published'], $form_state['values']['show_blocks'],
    $form_state['values']['did']);
    if ($form_state['values']['path'] != '' && module_exists('path')) {
      _og_panels_tab_path_alias($form_state['values']['nid'],
      $form_state['values']['tab_num'], $form_state['values']['path']);
    }
    drupal_set_message(t('Group page updated.'));
  }
  else {
    // Create a new display and record that.
    $display = panels_new_display();
    // TODO: let user select layout during creation.
    $display->layout = 'twocol_stacked';
    panels_save_display($display);
    $form_state['values']['tab_num'] = _og_panels_tab_num($form_state['values']['nid']);
    $sql = "INSERT INTO {og_panels} (did, nid, page_title, path, published, show_blocks, tab_num)
    VALUES (%d, %d, '%s', '%s', %d, %d, %d)";
    db_query($sql, $display->did, $form_state['values']['nid'], $form_state['values']['page_title'],
    $form_state['values']['path'], $form_state['values']['published'],
    $form_state['values']['show_blocks'], $form_state['values']['tab_num']);
    _og_panels_tab_renumber($form_state['values']['nid']);
    drupal_set_message(t('Group page created.'));
    $form_state['redirect'] = 'node/'. $form_state['values']['nid']. '/og_panels';
  }
}
```

# Form API Reference

[https://api.drupal.org/api/drupal/developer!topics!forms\\_api\\_reference.html/6](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/6)

API Homepage Log in / Register

## Form API Reference

### Form Controls

#### Legend:

X = attribute can be used with this type

- = this attribute is not applicable to this type

#type	checkbox	checkboxes	date	fieldset	file	password	password_confirm	radio	radios	select	textarea	textfield	weight
#access	X	X	X	X	X	X	X	X	X	X	X	X	X
#after_build	X	X	X	X	X	X	X	X	X	X	X	X	X
#ahah	X	-	-	-	-	X	-	X	-	X	X	X	-
#attributes	X	X	X	X	X	X	-	X	X	X	X	X	X
#autocomplete_path	-	-	-	-	-	-	-	-	-	-	-	X	-
#collapsed	-	-	-	X	-	-	-	-	-	-	-	-	-
#collapsible	-	-	-	X	-	-	-	-	-	-	-	-	-
#cols	-	-	-	-	-	-	-	-	-	-	X	-	-
#default_value	X	X	X	-	-	-	-	X	X	X	X	X	X
#delta	-	-	-	-	-	-	-	-	-	-	-	-	X
#description	X	X	X	X	X	X	X	X	X	X	X	X	X
#disabled	X	X	X	-	X	X	X	X	X	X	X	X	X
#element_validate	X	X	X	X	X	X	X	X	X	X	X	X	X
#field_prefix	-	-	-	-	-	-	-	-	-	-	-	X	-
#field_suffix	-	-	-	-	-	-	-	-	-	-	-	X	-
#maxlength	-	-	-	-	-	X	-	-	-	-	-	X	-
#multiple	-	-	-	-	-	-	-	-	-	X	-	-	-
#options	-	X	-	-	-	-	-	-	X	X	-	-	-
#parents	X	X	X	X	X	X	X	X	X	X	X	X	X
#post_render	X	X	X	X	X	X	X	X	X	X	X	X	X



# Traduzione contenuti

- Administer > Translate Interface

```
$items['admin/og/og'] = array(  
  'page callback' => 'drupal_get_form',  
  'page arguments' => array('og_admin_settings'),  
  'title' => 'Organic groups configuration',  
  'access arguments' => array('administer site configuration'),  
  'description' => 'Configure the main Organic groups module (og).',  
  'file' => 'og.admin.inc',  
  'file path' => drupal_get_path('module', 'og'). '/includes',  
  'weight' => -5,  
);
```

- Nelle traduzioni troviamo:
  - Titoli dei menù, blocchi, etc.
  - Stringhe scritte nel codice usando la funzione t():
  - ESEMPLI:
    - t('My Groups')
    - t("My Groups")

Overview Search Import Refresh Export

### TRANSLATE INTERFACE

Text group	String	Languages	Operations
Built-in interface	Organic groups configuration og.module:202;4235	ca zh-hans zh-hant cs da nl en-US et fi fr de e hu it ja lt pl pt sl es sv	<a href="#">edit</a> <a href="#">delete</a>

Search

String to be searched:  
Organic groups configuration  
Leave blank to show all strings. The search is case sensitive.

Language:  
All languages

Search in:

☒ Both translated and untranslated strings  
☐ Only translated strings  
☐ Only untranslated strings

Limit search to:

☒ All text groups  
☐ Built-in interface  
☐ Menu  
☐ Notifications  
☐ Blocks

Search

### EDIT STRING

Original text:  
Organic groups configuration

French:  
Configuration pour les groupes organiques

German:  
Konfiguration für organischen Gruppen

Italian:  
Configurazione per gruppi organici

English:  
Configuration for organic groups

=> Per permettere le traduzioni da amministrazione è quindi necessario ricorrere alla funzione t('stringa da tradurre')



# Altre funzioni

- `drupal_add_css(drupal_get_path('module', 'og'). '/theme/og.css');`
- `drupal_add_js(drupal_get_path('module', 'og'). '/og.js');`
- `variable_set('forum_default_name', "Default");`  
[https://api.drupal.org/api/drupal/includes!bootstrap.inc/function/variable\\_set/6](https://api.drupal.org/api/drupal/includes!bootstrap.inc/function/variable_set/6)
- `variable_get('forum_default_name', FALSE);`  
[https://api.drupal.org/api/drupal/includes!bootstrap.inc/function/variable\\_get/6](https://api.drupal.org/api/drupal/includes!bootstrap.inc/function/variable_get/6)
- `drupal_set_message($message = NULL, $type = 'status', $repeat = TRUE)`
  - [https://api.drupal.org/api/drupal/includes!bootstrap.inc/function/drupal\\_set\\_message/6](https://api.drupal.org/api/drupal/includes!bootstrap.inc/function/drupal_set_message/6)
  - ES: `drupal_set_message(t('Membership request approved.'));`
  - Con: `$type : {'status', 'warning', 'error'}`
- `drupal_mail_send($message);`
  - [https://api.drupal.org/api/drupal/includes!mail.inc/function/drupal\\_mail\\_send/6](https://api.drupal.org/api/drupal/includes!mail.inc/function/drupal_mail_send/6)
- `user_access('permission');`
- `l(t('Group Pages'), "node/$node->nid/og_panels");`
- `url("node/" . $nid . '/blog', array('absolute' => TRUE));`
  - <https://api.drupal.org/api/drupal/includes!common.inc/function/url/6>
- ...

# Link

- <http://drupal.org/>
- <http://www.drupalitalia.org/>
- [http://www.drupalitalia.org/sites/drupalitalia.org/files/Crazione\\_moduli\\_Drupal\\_0.pdf](http://www.drupalitalia.org/sites/drupalitalia.org/files/Crazione_moduli_Drupal_0.pdf)
- <http://api.drupal.org/api>

# Sistemi Distribuiti

## Corso di laurea in Ingegneria

**Programmare per il Web:  
Il linguaggio PHP**

- 1. Parte I: Introduzione**
- 2. Parte II: PHP**
  - **CMS**
- 3. Parte III: Approfondimenti**



# Espressioni regolari (1)

- Servono a definire la sintassi di sequenze di caratteri
- Si usano per trovare in una stringa la/le parti che hanno una sintassi particolare o per controllare se un testo ha la sintassi voluta
- Es:
  - Trovare in un testo tutte le email
  - Controllare se è stata inserita una email potenzialmente valida
  - Trovare in una pagina html tutti i link esterni (href=“...”)

# Espressioni Regolari (2)

- Una espressione regolare ha una specifica sintassi:
  - **'/pattern'**, dove **pattern** contiene una sequenza di caratteri, alcuni di questi possono essere caratteri speciali ( [], \, ., \*, ?, +, ^, \$, { } )
  - Es: /abc/ identifica i caratteri a,b e c in sequenza.
  - La / che indica inizio e fine del pattern può essere sostituita da un qualsiasi altro carattere (es. @ o |) basta che siano uguali, utile quando il carattere / fa parte del testo
- Semantica dei caratteri speciali:
  - [...] e [^...]
    - Indica il possibile valore di un carattere in un elenco (o non nell'elenco con [^...])
      - **[abc:]** indica il carattere a o b o c o :
      - **[a-z]** indica una qualsiasi lettera minuscola
      - **[0-9]** indica una qualsiasi cifra
      - **[a-zA-Z0-9]** indica una qualsiasi lettera o cifra
      - **[^0-9]** indica un carattere che non è una cifra
      - Es: '/0[1-9]0/' identifica le sequenze di tre cifre fatte da 0, una cifra non 0 e poi 0.

# Espressioni Regolari (3)

- Semantica caratteri speciali:
  - “.”
    - indica un qualsiasi carattere
      - Es: /a.c/ indica tre caratteri in cui la prima è a e ultima è c
  - “e\*”
    - Indica 0 o più occorrenze di **e** dove **e** è una espressione regolare o una lettera
      - **ab\*** indica il carattere a seguito da 0 o più b
      - **x[a-z]\*** indica il carattere x seguito da 0 o più lettere minuscole
      - **[1-9][0-9]\*** indica una cifra non 0 seguita da 0 o più cifre
      - **.\*** indica qualsiasi sequenza di caratteri
      - **a \*x** indica la lettera a seguita da 0 o più spazi seguiti da una x.



# Espressioni Regolari (4)

- Semantica caratteri speciali:
  - “ **$e^+$** ”
    - 1 o più occorrenze di  **$e$**  dove  **$e$**  è una espressione regolare o una lettera
      - **$ab^+$**  indica sequenze tipo  **$ab$** ,  **$abb$** ,  **$abbb$** , ...
  - “ **$e\{n,m\}$** ”
    - tra  **$n$**  e  **$m$**  occorrenze di  **$e$** 
      - **$[a-z]\{2,4\}$**  indica sequenze di 2, 3 o 4 lettere minuscole
  - “ **$e\{n,\}$** ”
    - almeno  **$n$**  occorrenze di  **$e$** 
      - **$[a-z]\{3,\}$**  indica sequenze di almeno 3 lettere minuscole
  - “ **$e\{,m\}$** ”
    - Al più  **$m$**  occorrenze di  **$e$** 
      - **$x[a-z]\{,3\}$**  indica  **$x$**  seguito da al massimo 3 lettere minuscole
  - “ **$e?$** ”
    - 0 o 1 occorrenza di  **$e$** 
      - **$-?[1-9][0-9]^*$**  indica un numero con un eventuale segno - iniziale

# Espressioni Regolari (5)

- Semantica caratteri speciali:
  - \...
    - Escape di un carattere speciale [,],.,\*,+... oppure sequenze predefinite
    - Es: \++ indica una sequenza di uno o più caratteri +
    - \d indica una cifra (equivalente a [0-9])
    - \D indica non una cifra
    - \s indica uno spazio
    - \S indica un non spazio
    - ...
  - “(...)”
    - Aggrega una sequenza di lettere e espressioni regolari
      - **a(b[0-9])+** indica una **a** seguita da una o più **b** seguita da una cifra (es: ab2b4b1)

# Espressioni Regolari (6)

- Semantica caratteri speciali:
  - “^...”
    - Indica che ciò che segue ^ deve essere all’inizio della stringa da controllare
      - Es: /^[0-9]/ indica che la stringa deve iniziare con una cifra
  - “...\$”
    - Indica che ciò che precede \$ deve essere alla fine della stringa da controllare
      - Es: /[0-9]\$/ indica che la stringa deve finire con una cifra
  - “^...\$”
    - Indica che ciò che sta tra ^ e \$ deve indicare completamente la stringa
      - Es: /^[ -\+]? \d+(\. \d\*)? \$/ indica che la stringa deve essere un numero decimale (es: 123, +23, -23.90, 0.99)



# Espressioni Regolari (7)

- Esempi:
  - Un IP (es: 150.217.15.241):  
**`/([0-9]{1,3}\.){3}[0-9]{1,3}/`**
  - Un MAC addr (es: 12:a2:23:aB:19:90)  
**`/([0-9a-fA-F]{2}:){5}[0-9a-fA-F]{2}`**
  - Una data (es: 12/02/2008)  
**`@([0-9]{1,2}/[0-9]{1,2}/[0-9]{4})@`**
  - Una e-mail (es: m.rossi@s.s-x.com)  
**`/[a-zA-Z\.\-0-9]+@[a-zA-Z0-9\.\-]+/`**

# Espressioni Regolari (7)

- Funzioni

- `int preg_match($pattern, $testo [$matches])`

- Ritorna 1 se il pattern fa match con il testo (0 altrimenti) e in `$matches` viene inserito la parte del testo che ha fatto match.

- Es:

`preg_match('/-?\d+/', 'abc -345 efg 23', $m)`

Ritorna 1 e `$m==array('-345')`

- `int preg_match_all($pattern, $testo, [$matches])`

- Ritorna il numero di parti del testo che fanno match con il pattern e in `$matches` vengono messi le parti del testo che fanno match.

- Es:

`preg_match_all('/-?\d+/', 'abc -345 efg 23', $m)`

Ritorna 2 e `$m==array(array('-345', '23'))`

# Espressioni Regolari (8)

- Funzioni:

- `string preg_replace($pattern, $replace, $testo)`

- Ritorna un nuovo testo dove tutte le occorrenze del pattern sono sostituite con il contenuto di `$replace`.

- Es:

```
preg_replace('/-?\d+/', 'N', 'abc -345 egf 23')
```

Ritorna "abc N egf N"

- Nella stringa `$replace` si possono indicare parti del pattern che ha fatto match usando `\1`, `\2`, `\3` ... che indicano le parti del pattern tra (...)

- Es:

```
preg_replace('/<(\d+)\.([a-z]*)>/', '(\2:\1)', 'abc<23.ab>defg<56.bf>')
```

Ritorna: "abc(ab:23)defg(bf:56)"

```
preg_replace('@([0-9]{1,2})/([0-9]{1,2})/([0-9]{4})@', '\3-\2-\1', "abc  
12/2/2004 cdef 26/8/2008 sed")
```

Ritorna: "abc 2004-2-12 cdef 2008-8-26 sed"



# Espressioni Regolari (9)

- Altre funzioni:
  - **preg\_grep**, ricerca elementi di array che fanno match con un pattern
  - **preg\_slice**, spezza una stringa in sottostringhe sulla base di un pattern

# Espressioni Regolari (10)

- Modificatori del pattern:
  - Alla fine di un pattern si può specificare una o più lettere che cambiano il comportamento del pattern: `'/[a-z]+/i'`
    - ***i*** : match case-insensitive
    - ***m*** : `^` e `$` fanno match anche dopo e prima di un a capo (multiline)
    - ***x*** : ignora gli spazi tra elementi del pattern
    - ***e*** : solo con `replace` interpreta la stringa da sostituire come codice PHP che produce la stringa da inserire
    - ...

# Espressioni Regolari (11)

- Match greedy (goloso)
    - Quando viene fatto il match di \* o + (specialmente se usati con . ) viene fatto il match con il numero maggiore possibile di caratteri.
    - Es:
      - pattern = @<a.\*>.\*</a>@
      - testo = 'abc<a href="aaa">defg</a>hilm<a>nopq</a>rst'
    - Per ovviare a questo si usa pattern:
      - @<a[^>]\*>[^<]\*</a>@
- oppure
- @<a.\*?>.\*?</a>@



# Cookie (1)

- I cookie permettono al server di memorizzare sul browser del client delle associazioni *chiave-valore* strettamente legate al server che le ha impostate.
- I cookie vengono inviati dal browser al server quando viene richiesta una pagina del server

# Cookie (2)

- In PHP per impostare un cookie si usa la funzione **setcookie(...)**
- Es:  
**setcookie("colore", "blu");**
  - Imposta sul browser che il colore preferito è il blu, questo cookie sarà attivo fino alla chiusura del browser
- Per accedere al contenuto di un cookie si usa l'array `$_COOKIE[nome]`
  - Es: `$_COOKIE["colore"]`
- **Attenzione**
  - i cookie vanno impostati prima del contenuto della pagina perchè fanno parte dello header del protocollo http
  - Quando sono impostati con `setcookie` il loro contenuto sarà disponibile in `$_COOKIE` solo alla prossima richiesta di una pagina web.

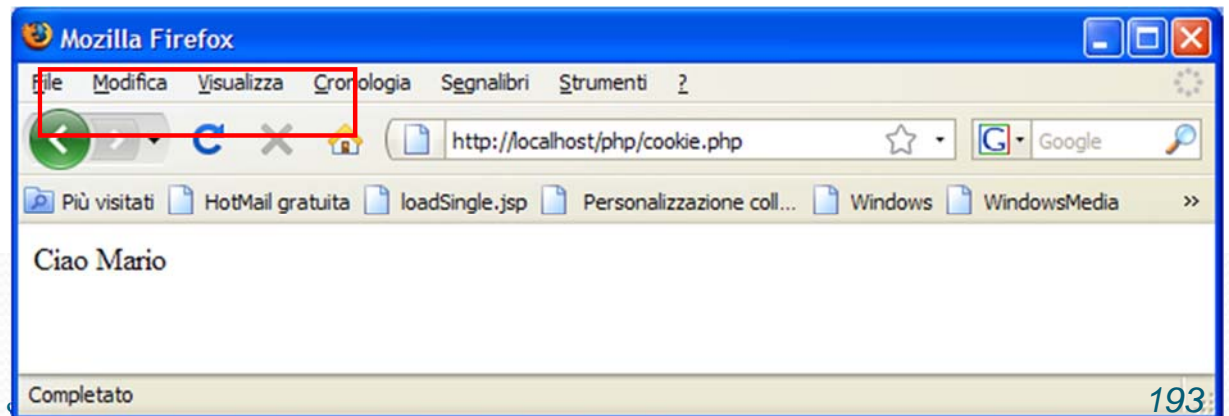
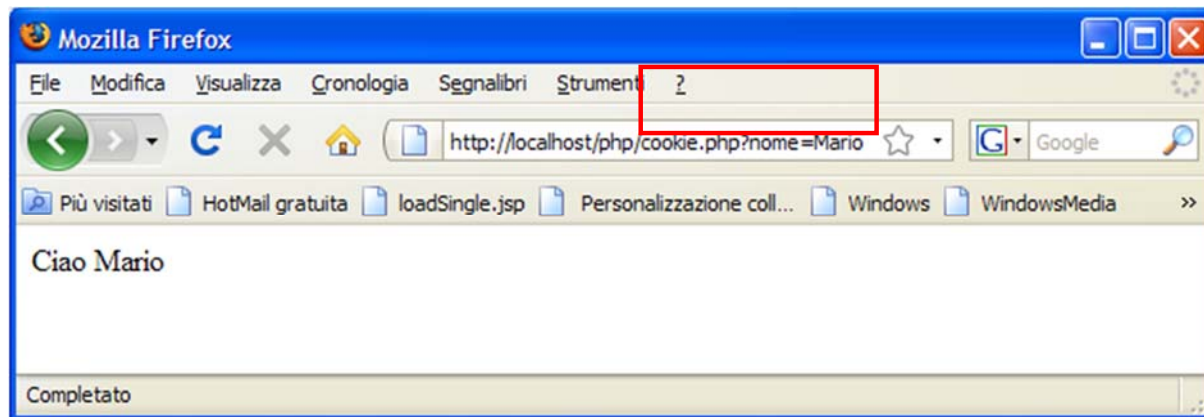
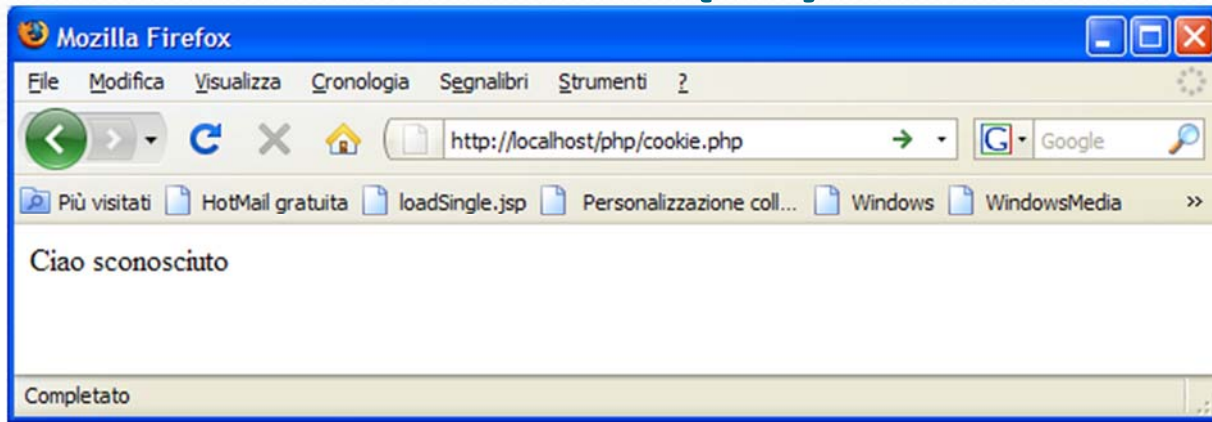
# Cookie (3)

## Esempio:

```
<?php
    //c'è un cookie con il nome?
    if(isset($_COOKIE["nome"]))
        $nome=$_COOKIE["nome"];
    else
        $nome="sconosciuto";
    //se il nome è passato nella url impostalo come cookie
    if(isset($_GET["nome"])) {
        $nome=$_GET["nome"];
        setcookie("nome", $nome);
    }
?>
<html>
    <body>
        <?php
            echo "Ciao $nome";
        ?>
    </body>
</html>
```



# Cookie (4)



# Cookie (5)

- Setcookie ha anche altri parametri:
  - `setcookie($name, $value, $expires=0, $path, $domain, $secure=false, $httponly=false)`
    - `$name` : nome del cookie
    - `$value` : valore associato
    - `$expires` : tempo in cui il cookie non sarà più valido, indicato in secondi da 1/1/1970 es: `time()+60*60` valido 1 ora, 0 indica valido fino a chiusura browser
    - `$path` : percorso sul server dove il cookie sarà accessibile “/” indica tutto il sito, per default la cartella dove si trova lo script
    - `$domain` : dominio dove il cookie sarà disponibile, es: “.example.com”
    - `$secure` : true se il cookie deve essere inviato solo in connessioni https
    - `$httponly` : true se il cookie può essere usato solo in connessioni http
- Per cancellare un cookie basta impostarlo al valore vuoto (attenzione che bisogna ridare tutti i parametri per identificare il cookie)

# Sessioni (1)

- Il protocollo *http* è **stateless**
- richieste diverse da una stessa sorgente sono viste come richieste completamente separate,
- questo non permette la realizzazione di funzionalità nelle quali è necessario individuare più richieste come facenti parte della stessa sequenza
- Es:
  - carrello spesa,
  - accesso controllato (login/logout)
  - contenuti personalizzati



# Sessioni (2)

- Per indicare che una pagina fa parte di una sessione si può usare:
  - IP del client tramite `$_SERVER["REMOTE_ADDR"]`
  - Un identificativo univoco della sessione passato in GET nella url o come campo hidden nei form passati in POST
  - Un identificativo univoco della sessione memorizzato come cookie nel browser
- La soluzione migliore è usare i cookie, o se i cookie non sono supportati o disabilitati usare GET/POST

# Sessioni (3)

- PHP fornisce delle funzionalità che permettono:
  - di indicare che le richieste fatte da un client fanno parte di una stessa sessione
  - di memorizzare informazioni associate alla sessione del client.
- PHP usa i cookie o le url nel caso i cookie siano disabilitati

# Sessioni (4)

- Per indicare che la pagina fa parte della sessione si usa la funzione **`session_start()`**, quindi si usa l'array **`$_SESSION[]`** per accedere e memorizzare i dati della sessione.
- **`session_start()`** identifica se una sessione è già attiva altrimenti genera un id sessione e lo imposta come cookie, carica **`$_SESSION`** con i dati della sessione presi da file su disco (sul server)
- Attenzione che **`session_start()`** deve essere chiamata prima di inviare il contenuto della pagina.



# Sessioni (5)

- Esempio:

```
<?php
    session_start();
?>
<html>
    <body>
        Count:
        <?php
            echo ++$_SESSION["count"];
        ?>
    </body>
</html>
```

# Sessioni (6)

- Nel caso i cookie non siano attivi, per mantenere la sessione quando si passa da una pagina all'altra si deve passare l'id della sessione in GET. Si può usare la costante SID (impostata da session\_start() ) per generare i link.

- Es:

```
echo "<a href='page.php?' . SID . ">pagina</a>"
```

- Genera link del tipo:

```
page.php?PHPSESSID=951f84d9115a4ac4a347cf2ee67a8907
```

# Sessioni (7)

- Per chiudere una sessione e cancellare i dati:

```
<?php
    session_start();
    $_SESSION=array();
    session_destroy();
?>
```



# Http headers

- Funzione ***header(string s)*** invia negli header della connessione http il contenuto della stringa s
- Es:

```
header('Location: http://server/test.php');
```

ridirige il browser all'indirizzo indicato

# Persistenza dei dati

- In moltissime applicazioni web dinamiche è necessario un supporto per la memorizzazione delle informazioni che:
  - Possa memorizzare un elevata quantità di informazioni
  - Sia efficiente nel reperimento delle informazioni necessarie
  - Supporti l'utilizzo concorrente
  - Sia basato su standard industriali

# Basi di Dati & SQL



# Basi di Dati

- Le **basi di dati**, o **Data Base**, sono sistemi che permettono la manipolazione di grandi quantità di informazioni.
- Consentono infatti:
  - Raccolta e acquisizione dei dati
  - L'archiviazione e la conservazione
  - La loro elaborazione e trasformazione
  - La distribuzione, comunicazione e scambio dell'informazione
- Il **Data Base** è l'insieme dei dati che organizzati costituiscono l'informazione.
- Il data base, però, non è costituito solo da un insieme di dati
  - **tabelle** (contenitori di dati)
  - **query** (strumenti di interrogazione, recupero dati),
  - **maschere** (strumenti per l'inserimento su computer)
  - **report** (output dei dati).

# Sistemi di gestione di basi dati

- Un sistema di gestione di basi di dati (DataBase Management System, DBMS) è un sistema software in grado di gestire collezioni di dati che siano grandi, condivise, e persistenti, assicurando la loro affidabilità e privatezza. Un DBMS deve essere efficiente ed efficace.
- In seconda approssimazione, si può definire una base di dati come una

“collezione di dati gestita da un DBMS”.

# Sistemi di gestione di basi dati

- Le basi di dati possono essere **grandi**:
  - IBM certifica basi di dati di oltre 64 Gb!
  - basi di dati gestite da DBMS che devono prevedere gestione di dati in memoria di massa.
- Le basi di dati sono **condivise**:
  - applicazioni e utenti diversi devono poter accedere, secondo opportune modalità, a dati comuni
  - riduzione di ridondanze e di inconsistenze.
  - Per garantire l'accesso condiviso ai dati, ogni DBMS dispone di un meccanismo di controllo di concorrenza.



# Condivisione

- In una organizzazione ogni componente è interessata ad una porzione del Sistema Informativo
- Queste porzioni possono sovrapporsi
- Una base di dati è una risorsa integrata condivisa da diverse componenti
- L'integrazione e la condivisione permettono di ridurre ridondanze (dati parzialmente o totalmente replicati) e conseguenti inconsistenze

# Condivisione

- La condivisione non è mai completa: controllo della privacy e regolamentazione degli accessi
- La condivisione comporta la necessità di gestire accessi contemporanei agli stessi dati: controllo della concorrenza

# Sistemi di gestione di basi dati

- Le basi di dati sono **persistenti**,
  - ovvero hanno un tempo di vita che non è limitato a quello delle singole esecuzioni dei programmi che le utilizzano.
  - I dati non cessano di esistere con il termine dei programmi che li elaborano.
- I DBMS garantiscono l'**affidabilità**,
  - cioè la capacità di conservare intatto il contenuto della base di dati in caso di malfunzionamenti hardware e/o software.
  - Backup e recovery
- I DBMS garantiscono la **privatezza dei dati**:
  - ciascun utente autenticato viene abilitato a svolgere solo certe azioni sui dati.



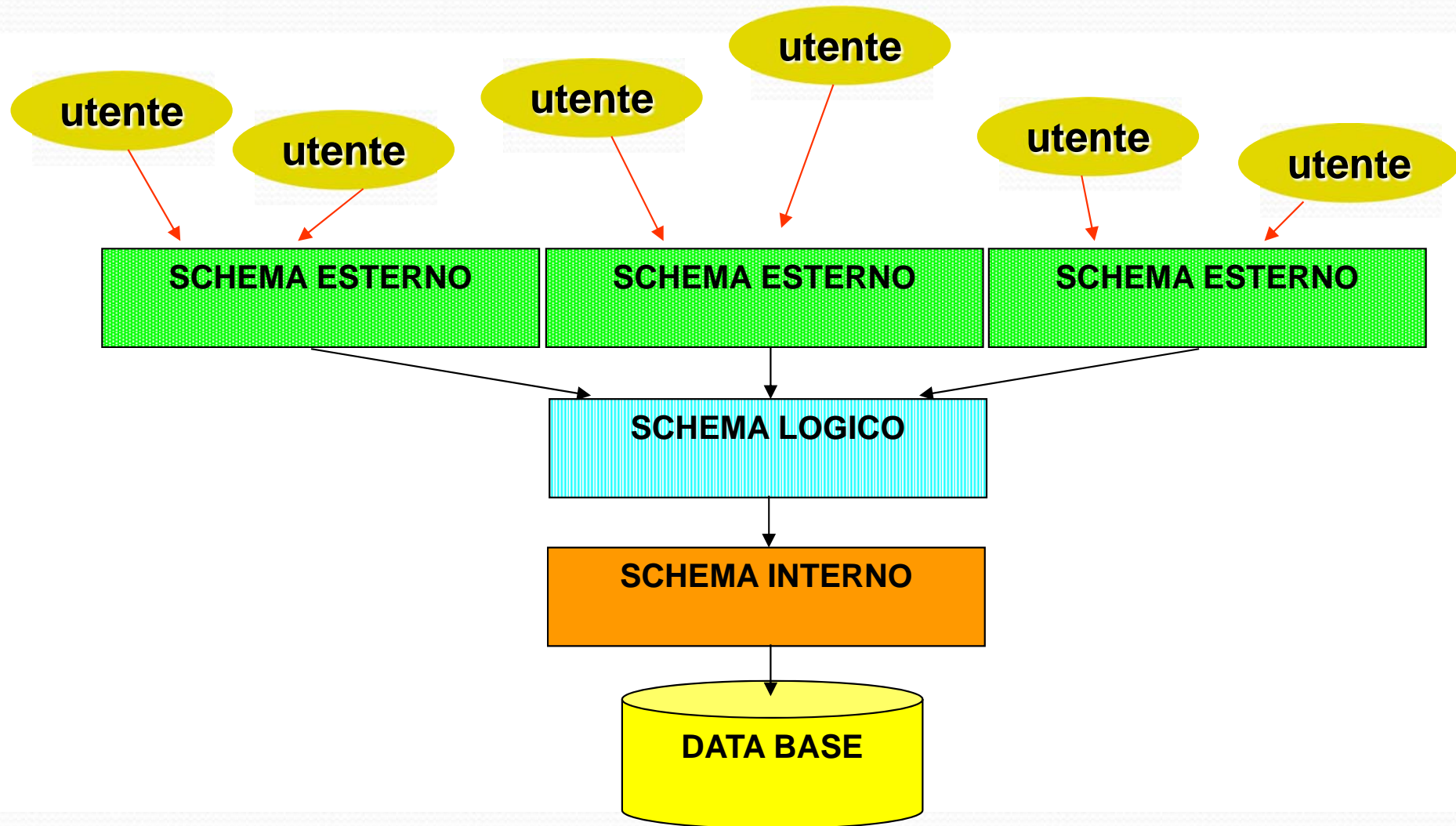
# Memorizzazione

- I sistemi di basi di dati utilizzano file in formati proprietari per memorizzare i dati ma
- offrono agli utenti una vista astratta dei dati, in modo da rendere trasparenti i dettagli di memorizzazione e manipolazione

# Sistemi di gestione di basi dati

- Per **efficienza** di un DBMS si intende
  - la sua capacità di svolgere le operazioni usando un insieme di risorse che siano accettabili per gli utenti.
  - tempo e spazio
- Per **efficacia** di un DBMS si intende
  - la capacità di rendere produttive le attività dei suoi utenti.

# I 3 livelli di astrazione di un DB





# I 3 livelli di astrazione di un DB

- **Schema esterno**

- descrizione di una porzione della base di dati in un modello logico attraverso “viste” parziali, o derivate, che possono prevedere organizzazioni dei dati diverse rispetto a quelle utilizzate nello schema logico, e che riflettono esigenze e privilegi di accesso di particolari tipologie di utenti; ad uno schema logico si possono associare più schemi esterni

- **Schema logico**

- descrizione dell'intera base di dati nel modello logico “principale” del DBMS, ad esempio la struttura delle tabelle

- **Schema fisico**

- rappresentazione dello schema logico per mezzo di strutture fisiche di memorizzazione, cioè i file

# Una vista (schema esterno)

## Corsi

Corso	Docente	Aula
Basi di dati	Rossi	DS3
Sistemi	Neri	N3
Reti	Bruni	N3
Controlli	Bruni	G

## Aule

Nome	Edificio	Piano
DS1	OMI	Terra
N3	OMI	Terra
G	Pincherle	Primo

## SCHEMA LOGICO

## CorsiSedi

Corso	Aula	Edificio	Piano
Sistemi	N3	OMI	Terra
Reti	N3	OMI	Terra
Controlli	G	Pincherle	Primo

**VISTA**

# Indipendenza tra Dati e Accesso

- Gli accessi alla base di dati avvengono solamente attraverso lo schema esterno, che può coincidere completamente con lo schema logico.

- **Indipendenza fisica**

il livello logico e quello esterno sono indipendenti da quello fisico

- una relazione è utilizzata nello stesso modo qualunque sia la sua realizzazione fisica (organizzazione dei file e loro allocazione fisica)
- la realizzazione fisica può cambiare senza che debbano essere modificati i programmi



# Schemi e istanze

- In ogni base di dati esistono:
  - lo schema, sostanzialmente invariante nel tempo, che ne descrive la struttura (aspetto intensionale): nel modello relazionale, le intestazioni delle tabelle = lista di nomi di attributi e loro tipi
  - l'istanza, i valori attuali, che possono cambiare anche molto rapidamente (aspetto estensionale): nel modello relazionale, il “corpo” di ciascuna tabella

# Esempio: ...nel modello relazionale

## ANAGRAFICA

NOME	COGNOME	DATAN	LUOGON
Piero	Napoli	22-10-63	Bari
Marco	Bianchi	01-05-54	Roma
Maria	Rossi	09-02-68	Milano
Maria	Bianchi	07-12-70	Bari
Paolo	Sossi	15-03-75	Palermo

## SCHEMA

## ISTANZA

# Linguaggi per le basi di dati

- **data definition language (DDL)**
  - per la definizione di **schemi** (logici, esterni, fisici) e altre operazioni generali
- **data manipulation language (DML)**
  - per l'interrogazione e l'aggiornamento di (**istanze** di) basi di dati
- Nei DBMS Standard
  - **SQL** integra in un unico linguaggio i due tipi di funzionalità



# Cenni storici

**1970:** E.F. Codd (IBM) introduce il **modello relazionale**

**anni '70:** prototipi sistemi relazionali (System R, IBM)

**anni '80:** sistemi relazionali commerciali (Ingress, Oracle, ...)

**anni '90:** sistemi relazionali personali  
(access) e open source (mySQL, postgres)


**anni '00:** sistemi relazionali embedded  
(SQLite)

# Modelli di dati

Il **modello relazionale dei dati** permette di definire nuovi “*concetti*” per mezzo del *meccanismo di strutturazione* noto come “relazione” che consente di organizzare i dati in insiemi di *aggregazioni* (record) a struttura fissa.

Una relazione si rappresenta per mezzo di una **tabella**, le cui righe rappresentano specifici record.

# Modello relazionale

- Oggetto = Record
- Campi = Informazioni  interesse
- Oggetto = “Membro dello Staff”
- Informazioni di interesse = Codice, Cognome, Nome, Ruolo, Anno di assunzione

CODICE	COGNOME	NOME	RUOLO	ASSUNZIONE
COD1	Rossi	Mario	Analista	1995



# Modello relazionale

- Tabella = Insieme di record di tipo omogeneo
- Tabella STAFF = Insieme di record di tipo “Membro dello Staff”



CODICE	COGNOME	NOME	RUOLO	ASSUNZIONE
COD1	Rossi	Mario	Analista	1995
COD2	Bianchi	Pietro	Analista	1990
COD3	Neri	Paolo	Amministratore	1985

# Modelli di dati

## Esempio:

posso organizzare i dati relativi ai docenti e ai relativi corsi universitari per mezzo della relazione “docenza” rappresentabile dalla seguente tabella

## DOCENZA

<i>Corso</i>	<i>Nome_docente</i>
Informatica	Bianchi
Matematica	Rossi
Diritto	Verdi
Storia	Pinco
Statistica	Pallino

# Esempio di un DB relazionale

## STUDENTI

Matric	Cognome	Nome	DataNasci
276545	Smith	Mary	25/11/1980
485745	Black	Anna	23/04/1981
200768	Verdi	Paolo	12/02/1981
587614	Smith	Lucy	10/10/1980
937653	Brown	Mavis	01/12/1980

## ESAMI

### CORSI

Codice	Titolo	Tutor	Stud	Voto	Corso
01	Physics	Grant	276545	C	01
03	Chemistry	Beale	276545	B	04
04	Chemistry	Clark	937653	B	01
			200768	B	04



# Esempio di DB relazionale

## STUDENTI

Matric	Cognome	Nome	DataNasci
276545	Smith	Mary	25/11/1980
485745	Black	Anna	23/04/1981
200768	Verdi	Paolo	12/02/1981
587614	Smith	Lucy	10/10/1980
937653	Brown	Mavis	01/12/1980

## CORSI

Codice	Titolo	Tutor
01	Physics	Grant
03	Chemistry	Beale
04	Chemistry	Clark

## ESAMI = relazione tra studenti e corsi

Stud	Voto	Corso
276545	C	01
276545	B	04
937653	B	01
200768	B	04

# Schemi e istanze di relazioni

Per **schema di una relazione** si intende la sua intestazione cioè il nome della relazione seguito dai nomi dei suoi attributi; esempio:

DOCENZA(Corso, Nome\_docente).

L'**istanza di una relazione** è costituita dall'insieme delle sue righe.

Lo schema di relazione è stabile nel tempo, a differenza della sua istanza.

# Schemi e istanze di relazioni

DOCENZA

<i>Corso</i>	<i>Nome_docente</i>
Informatica	Bianchi
Matematica	Rossi
Diritto	Verdi
Storia	Pinco
Statistica	Pallino

Istanza della relazione

Schema della relazione

Record



# Relazioni e basi dati

Una base di dati è generalmente costituita da più relazioni, i cui record contengono valori comuni, ove necessario per stabilire **corrispondenze** (*associazioni*).

# Relazioni e basi dati

## STUDENTI

<i>Matricola</i>	<i>Cognome</i>	<i>Nome</i>	<i>Data_nascita</i>
389726	Rossi	Maria	25/07/1981
421695	Bianchi	Antonio	11/03/1978
379723	Verdi	Luca	03/09/1980

## ESAMI

<i>Studente</i>	<i>Voto</i>	<i>Corso</i>
389726	28	01
389726	21	04
379723	25	04
421695	26	03

## CORSI

<i>Codice</i>	<i>Titolo</i>	<i>Docente</i>
01	Informatica	Pinco
02	Matematica	Pallino
03	Storia	Bello
04	Statistica	Alto

# Relazioni e basi dati

L'esempio precedente mostra una delle caratteristiche fondamentali del modello relazionale: i riferimenti fra dati in relazioni diverse sono rappresentati per mezzo di valori. Per tale motivo spesso si afferma che: “il modello relazionale è basato sui valori”



# Integrità

I dati devono soddisfare dei “vincoli” che esistono nella realtà di interesse

- uno studente risiede in una sola città (**dipendenze funzionali**)
- la matricola identifica univocamente uno studente (**vincoli di chiave**)
- un voto è un intero positivo compreso tra 18 e 30 (**vincoli di dominio**)
- lo straordinario di un impiegato è dato dal prodotto del numero di ore per la paga oraria
- lo stipendio di un impiegato non può diminuire (**vincoli dinamici**)

# Informazione incompleta e valori nulli

A volte può accadere che non si dispongano di tutti i dati di una *tupla* (record) di una relazione. **Esempio:** per uno studente potrei non disporre del numero di matricola o della data di nascita. Come fare?

# Informazione incompleta e valori nulli

Per rappresentare in modo semplice la non disponibilità di valori, il concetto di relazione viene esteso prevedendo che un record possa assumere su ogni suo campo, un valore speciale (detto **valore nullo**) che denota l'assenza di informazione.



# Informazione incompleta e valori nulli

## Nota bene:

È opportuno controllare la presenza di valori nulli nelle relazioni perché solo alcune configurazioni possono essere ammesse

## STUDENTI

<i><b>Matricola</b></i>	<i><b>Cognome</b></i>	<i><b>Nome</b></i>	<i><b>Data_nascita</b></i>
389726	Rossi	Maria	NULL
NULL	Bianchi	Antonio	11/03/1978
...	...	...	...

# Informazione incompleta e valori nulli

Un valore nullo per la matricola di uno studente comporta la perdita dei legami fra record di tabelle diverse.

(*record* è sinonimo di *tupla*; *tabella* è sinonimo di *relazione*)

## STUDENTI

<b>Matricola</b>	<b>Cognome</b>	<b>Nome</b>	<b>Data_nascita</b>
389726	Rossi	Maria	NULL
NULL	Bianchi	Antonio	11/03/1978
...	...	...	...

# Informazione incompleta e valori nulli

Quando si definisce una relazione è possibile specificare che i valori nulli sono ammessi solo su alcuni attributi e non su altri.



# Vincoli di integrità

In alcuni casi può accadere che, a seguito di errori umani nell'introduzione di dati, alcuni record delle tabelle rappresentino informazioni non corrette.

# Vincoli di integrità

In una base di dati è opportuno evitare situazioni come questa. A tale scopo è stato introdotto il concetto di *vincolo di integrità* come *proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette*.

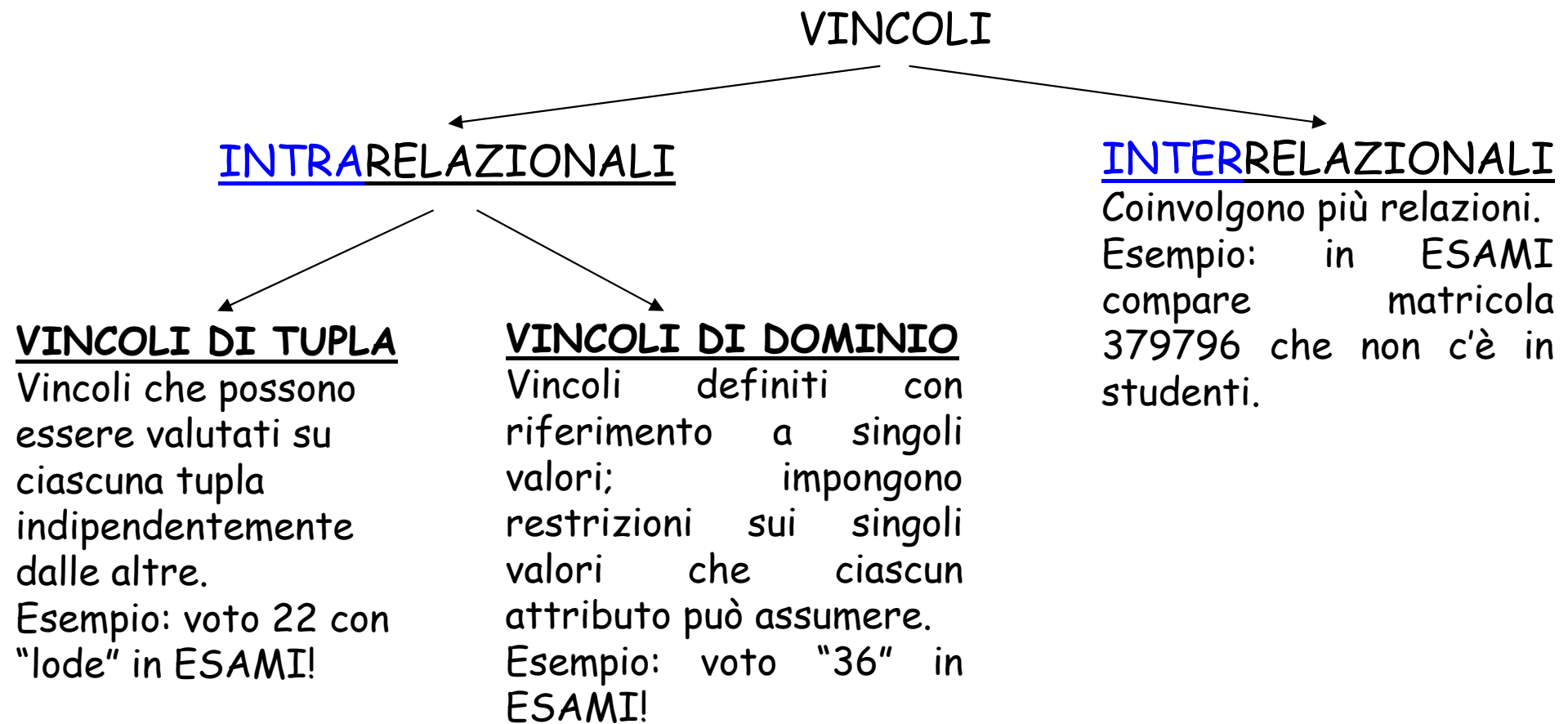
## STUDENTI

<b>Matricola</b>	<b>Cognome</b>	<b>Nome</b>	<b>Data_nascita</b>
200768	Verdi	Fabio	<u>12/02/1799</u>
942761	Rossi	Luca	10/10/1979
<u>942761</u>	Bianchi	Mario	14/07/1983

## ESAMI

<b>Studente</b>	<b>Voto</b>	<b>Lode</b>	<b>Corso</b>
942761	<u>22</u>	<u>Si</u>	01
942761	<u>36</u>	No	03
<u>379796</u>	30	No	01

# Vincoli di integrità





# Chiavi

I più importanti vincoli intrarelazionali sono i **vincoli di chiave**: senza di essi il modello relazionale perderebbe la sua utilità e quindi il suo senso! **Una chiave è un insieme di attributi utilizzati per identificare univocamente i record di una relazione.** (esempi)

# Chiavi

Come caso estremo una chiave potrebbe essere costituita da tutti gli attributi di una relazione; di solito si usano chiavi minimali ovvero composte dal minor numero di attributi. Esempi: il **numero di matricola** di uno studente o il **codice di un corso** o il **codice fiscale** che ci identifica univocamente.

# Chiavi

Il fatto che su ciascuno schema di relazione possa essere definita almeno una chiave **garantisce l'accessibilità di tutti i valori** di una base di dati e la loro univoca identificabilità.

**(posso ricercare, modificare e/o eliminare le tuple)**



# Chiavi

Inoltre, il vincolo di chiave **permette di stabilire efficacemente le corrispondenze fra dati contenuti in relazioni diverse** che caratterizzano il modello razionale come “*modello basato sui valori*”.

# Chiavi

NOTA BENE:

Bisogna limitare il più possibile la presenza di valori nulli nelle chiavi. Su una delle chiavi minimali detta **chiave primaria** è assolutamente vietata la presenza di valori **nulli**.

# Chiave primaria

Insieme minimale di attributi di una relazione che consente di identificare univocamente i record di una tabella; non può ammettere valori nulli.



# Chiavi

In quasi tutti i casi è possibile trovare delle chiavi minimali e quindi scegliere una chiave primaria. Nei rari casi in cui ciò non accada è necessario introdurre **un attributo aggiuntivo** (Un codice. Probabilmente non significativo al fine interpretativo dei dati) i cui valori vengano generati automaticamente e associati a ciascuna tupla della relazione.

# Vincoli di integrità referenziale

## AUTO

<u>Targa</u>	<u>Cognome</u>	<u>Nome</u>
BK749L	Rossi	Mino
CM294P	Bianchi	Tino
RL296K	Verdi	Pino
WR692Y	Celesti	Lino

Le informazioni della relazione INFRAZIONI sono rese significative attraverso il riferimento alla relazione AUTO. Se un valore di "Codice\_targa" in infrazioni non compare come valore di "Targa" in AUTO, il riferimento non è efficace.

## INFRAZIONI

<u>Codice</u>	<u>Data</u>	<u>Articolo</u>	<u>Codice_targa</u>
743792	26/10/2004	34	BK749L
927450	10/10/2004	53	RL296K
992001	04/10/2004	15	WR692Y
999223	10/09/2004	27	BK749L

# Vincoli di integrità referenziale

Un vincolo di integrità referenziale (detto anche chiave esterna) tra un insieme  $X$  di attributi di una relazione  $R_1$  e un'altra relazione  $R_2$  è soddisfatto se i valori su  $X$  di ciascuna tupla dell'istanza di  $R_2$  compaiono come valori della chiave primaria dell'istanza di  $R_1$ .



# Transazione

**Transazione:** sequenza di operazioni che costituiscono un'unica operazione logica

“Trasferisci €1000 dal c/c c1 al c/c c2”

cerca c1

modifica *saldo* in *saldo*-1000

cerca c2

modifica *saldo* in *saldo*+1000

Una transazione deve essere eseguita completamente (**committed**) o non deve essere eseguita affatto (**rolled back**)

# Concorrenza

**Transazione 1:** “Accredita €1000 sul c/c c1”

**Transazione 2:** “Accredita €500 sul c/c c1”

Transazione 1	Tempo	Transazione 2
cerca c1	t1	
	t2	cerca c1
modifica <i>saldo</i> in <i>saldo</i> +1000	t3	
	t4	modifica <i>saldo</i> in <i>saldo</i> +500

**valore iniziale *saldo*:** 2000

**valore finale *saldo*:** 2500

# Relazioni

- Come modellare le relazioni tra elementi delle tabelle?
- Tipi di relazione
  - Uno a uno (1:1)
  - Uno a molti (1:n)
  - Molti a molti (m:n)
  - Autoreferenzianti
  - Relazioni opzionali



# Relazioni Uno a Uno

- Due tabelle A e B sono in **relazione uno a uno** se per una singola riga della tabella A esiste solo una riga della tabella B ad essa relazionata e viceversa
- **Esempio:** La relazione tra due tabelle che sono l'una sottoinsieme dell'altra è sempre di tipo uno a uno
- La tabella “Persone” è in relazione uno a uno con la tabella “Studente” e con la tabella “Lavoratori”
  - La tabella “Persone” viene detta *tabella principale* e riporta informazioni generiche sulle persone;
  - Le tabelle “Studenti” e “Lavoratori” vengono dette *tabelle subordinate* e riportano informazioni più specifiche
- Persone(**IdPersona**, Nome, Cognome, DataNasc)
- Studenti(**IdPersona**, Matricola)
- Lavoratori(**IdPersona**, Professione)

# Relazioni Uno a Uno

- Implementazione:
  - Nella tabella subordinata si inserisce un campo nel quale si mette la chiave primaria della tabella principale
  - Questo campo può essere chiave primaria della tabella subordinata (es: relazione di sottoinsieme)
- Esempio:
  - Persone(**IdPersona**, Nome, Cognome, DataNasc)
  - Studente(**IdPersona**, Matricola)
  - Lavoratore(**IdPersona**, Professione)

# Relazioni Uno a Molti

- Due tabelle A e B sono in relazione uno a molti se ogni riga della tabella A ammette da zero a n righe della tabella B ad essa relazionate.
- Inoltre, ogni riga della tabella B ammette una sola riga della tabella A ad essa relazionata.
- **Esempio:** La tabella “CorsiDiLaurea” è in relazione uno a molti con la tabella “Studenti”.
  - La tabella “CorsiDiLaurea” è la *tabella principale*;
  - La cardinalità della relazione è di tipo uno a molti perché ogni studente è iscritto (in questo esempio) ad un solo corso di laurea, mentre ogni corso di laurea ammette più studenti iscritti



# Relazioni Uno a Molti

- Implementazione:
  - si identificano la tabella principale e quella secondaria;
    - la direzione della relazione vede il simbolo di cardinalità “uno” dal lato della tabella principale ed il simbolo “molti” dal lato della tabella secondaria;
  - si duplica la chiave primaria della tabella principale nella tabella secondaria.
- Esempio:
  - CorsiDiLaurea(idCDL, Nome)
  - Studenti(Matricola, Nome, Cognome, **idCDL**)

# Relazioni Molti a Molti

- Due tabelle A e B sono in relazione molti a molti quando ogni riga della tabella A può essere legata a molte righe della tabella B.
- Inoltre, ogni riga della tabella B può essere legata a molte righe della tabella A.
- **Esempio:**
  - Le tabelle “Studenti” e “Corsi” stabiliscono una relazione molti a molti visto che:
    - ogni studente può frequentare più corsi e
    - ogni corso può essere frequentato da molti studenti.

# Relazioni Molti a Molti

- Implementazione
  - si crea una nuova tabella definita come “tabella di relazione”;
  - la chiave primaria della “tabella di relazione” è composta dalle due chiavi primarie delle tabelle coinvolte;
  - si stabilisce una relazione uno a molti tra ognuna delle due tabelle e la “tabella di relazione”.
- Esempio
  - Studenti(Matricola, Nome, Cognome)
  - Corsi(IdCorso, NomeCorso)
  - StudentiCorsi(Matricola, IdCorso)



# Relazioni Autoreferenzianti

- Esistono relazioni autoreferenzianti cioè tra righe della stessa tabella
- Possono essere relazioni 1:1, 1:n, m:n autoreferenzianti
- Si trattano come le relazioni viste precedentemente solo che tabelle principale e subordinata coincidono.
- Esempio: Legame genitore/figli tra persone

# Relazioni Opzionali

- Esistono casi nelle relazioni 1:1, 1:n e m:n in cui in realtà si può anche non avere relazioni con elementi dell'altra tabella (1:0..1)
- In questi casi si può usare il valore NULL in un campo usato per legare una tabella ad un'altra per indicare che non è in relazione con nessun elemento.



# SQL

## (Structured Query Language)

Un linguaggio per DBMS  
relazionali



# Introduzione

- Nasce nella seconda metà degli anni Settanta presso i laboratori di ricerca della IBM
- Viene standardizzato dagli organismi ANSI e ISO, con la partecipazione di gran parte di produttori di DBMS, nel 1986/1989 (“SQL/89”): supporto delle funzionalità di DML
- Viene proposta una seconda versione nel 1992 (“SQL/92” o “SQL2”), con l’aggiunta di funzionalità di DDL. Vengono proposti vari livelli di funzionalità:
  - *Easy* (molto simile a SQL89),
  - *Intermediate* (viene garantito dalla maggior parte dei produttori),
  - *Full* (non viene supportato da tutti, ma ciascun produttore inserisce estensioni proprietarie al linguaggio).
- Ultimo standard: SQL2008

# Creazione Tabella

- Tabella: collezione di attributi e un insieme di vincoli (opzionale).
- Es. di creazione della tabella “Dipartimento”:

```
CREATE TABLE Dipartimento  
(  
    Nome char(20) primary key,  
    Indirizzo char(50),  
    Citta char(20)  
)
```

# Creazione Tabella

Sintassi:

```
Create table NomeTabella  
(  
  Attributo1 Tipo [modifTipo] [valoreDiDefault] [Vincoli],  
  Attributo2 Tipo [modifTipo] [valoreDiDefault] [Vincoli],  
  ...,  
  [AltriVincoli]  
)
```

Crea una tabella di nome “NomeTabella” avente come attributi Attributo1, Attributo2, ... AttributoN, ciascuno dei quali definito nell’ambito di un tipo o dominio (insieme dei possibili valori attribuibili all’attributo) con eventualmente indicati alcuni modificatori di tipo (Es: unsigned), preimpostato a un valore predefinito (“valore di default”) ed eventualmente soggetto a un insieme di vincoli.



# Creazione Tabella

Tre concetti da analizzare in dettaglio:

- Tipi/Domini
- Valori di default degli attributi
- Vincoli

# Tipi/Domini

SQL mette a disposizione quattro famiglie di Tipi/Domini elementari:

- Caratteri
- Numeri esatti
- Numeri approssimati
- Data e ora

# Tipi/Domini

**Carattere:** permette di rappresentare singoli caratteri o stringhe alfanumeriche (sequenze di caratteri composte da lettere, numeri e altri simboli; note anche come “stringhe”).

Si divide in:

- **Lunghezza fissa.** Es, **CHAR(20)**: il sistema riserva sempre 20 caratteri per l'attributo.
- **Lunghezza variabile.** Es, **VARCHAR(20)**: il sistema riserva il numero di caratteri indispensabile a rappresentare la stringa e comunque non più di 20.



# Tipi/Domini

**Numeri esatti:** usato per rappresentare valori interi o in virgola fissa. Il sistema non commette errori di rappresentazione del dato.

- **INTEGER:** interi “grandi” (4 byte)
- **TINYINT:** interi “molto piccoli” (1 byte)
- **SMALLINT:** interi “piccoli” (2 byte)
- **BIGINT:** interi “molto grandi” (8 byte)
- **DECIMAL(precisione, scala):** numeri in virgola fissa – precisione = numero di cifre complessive; scala = numero di cifre parte decimale
- **NUMERIC(precisione, scala):** analogo a DECIMAL, ma i parametri indicati rappresentano un requisito minimo per il sistema, che può quindi scegliere una rappresentazione interna più accurata.

Es: **DECIMAL(6,3)** va da -999,999 a +999,999 (=6 cifre totali, di cui 3 riservate per la parte frazionaria)

# Tipi/Domini

**Numeri approssimati:** numeri rappresentati internamente come (mantissa normalizzata, esponente): non possono rappresentare *tutti* i numeri, quindi gli altri vengono approssimati ai più vicini.

- **REAL:** numeri reali a precisione più bassa
- **DOUBLE PRECISION:** numeri reali a precisione doppia rispetto a REAL
- **FLOAT(precisione):** numeri reali con prefissato numero di bit per la mantissa

# Tipi/Domini

- **Data e ora:** rappresentano istanti di tempo
  - **DATE:** solo data
  - **TIME:** solo ora
  - **DATETIME:** data e ora
  - **TIMESTAMP:** data e ora da 1/1/1970



# Valori di default

Il valore di default viene usato per fornire un valore predefinito ad un attributo che non sia specificato al momento dell'inserimento di una nuova riga nella tabella.

Con la seguente definizione di tabella, ad esempio,

```
CREATE TABLE Persona (  
Nome varchar(20),  
Figli smallint default 0  
)
```

**INSERT INTO Persona (Nome) VALUES 'Verdi Giuseppe'**  
inserirà una riga composta da “Verdi Giuseppe” come nome e 0 come figli

# Vincoli

- Due tipi di vincoli:
  - **Intra**relazionali (che coinvolgono una sola tabella)
  - **Inter**relazionali (che sono espressi su più tabelle)
- I vincoli imposti vengono sempre verificati ad ogni operazione dal DBMS, che quindi la rifiuterà in caso di loro violazione.
- Verranno esposti i tipi di vincoli *elementari* per entrambi i casi.

# Vincoli intrarelazionali

- **Not null:** impone ad un attributo di non poter assumere mai il valore nullo.

Es:

```
CREATE TABLE ... (  
...  
Cognome CHAR(20) not null  
...  
)
```



# Vincoli intrarelazionali

- **Unique:** Si applica ad un attributo o ad un insieme di attributi per imporre che i valori siano unici, cioè non esistano righe distinte con valori uguali.
- **Es:**  
    **Matricola char(6) unique**  
    Impone che non esistano due righe con lo stesso numero di matricola.
- E' possibile anche indicare una coppia di valori:  
    **UNIQUE(Cognome, Nome)**

# Vincoli intrarelazionali

- Es (a):  
**Nome CHAR(20) NOT NULL,**  
**Cognome CHAR(20) NOT NULL,**  
**UNIQUE (Cognome, Nome)**
- Es (b):  
**Nome CHAR(20) NOT NULL UNIQUE,**  
**Cognome CHAR(20) NOT NULL UNIQUE**
- L'esempio (a) impone che non esistano due righe che abbiano *contemporaneamente* Cognome e Nome uguali, mentre il secondo esempio non ammette duplicati di riga su valori dei singoli campi (=> il caso (b) è più restrittivo).

# Vincoli intrarelazionali

- **Primary key:** definisce un attributo (o un insieme di attributi) come chiave primaria della tabella.
- Es:  
`Nome CHAR(20),  
Cognome CHAR(20),  
Dipart CHAR(15),  
Stipendio NUMERIC(9) default 0,  
PRIMARY KEY(Cognome, Nome)`
- Es:  
`Matricola CHAR(6) PRIMARY KEY,  
Cognome CHAR(20),  
Nome CHAR(20)`
- Nota: PRIMARY KEY può essere usato solo una volta nella definizione di una tabella!



# Vincoli interrelazionali

- **references:** definisce l'attributo di un'altra tabella col quale l'attributo in corso di definizione deve essere legato
- Es:

```
CREATE TABLE Persona(  
    CodFiscale CHAR(16) PRIMARY KEY,  
    Cognome CHAR(20),  
    Nome CHAR(20),  
    NomeCitta CHAR(15) REFERENCES Citta(Nome)  
)
```

# Vincoli interrelazionali

- **Foreign key**: analogo a **references**, ma usato per insiemi di attributi
- Es:

```
CREATE TABLE Impiegato(  
    Matricola char(6) primary key,  
    Nome varchar(20) not null,  
    Cognome varchar(20) not null,  
    unique (Cognome, Nome),  
    foreign key (Nome, Cognome)  
        references Anagrafica(Nome, Cognome)  
)
```

# Politiche di risposta del DBMS su violazione vincoli di integrità referenziale

- Su violazione in seguito a cancellazione o aggiornamento è possibile imporre:
  - **Cascade**: Applicare in cascata ai referenzianti
  - **Set Null**: Impostare i referenzianti a null
  - **Set Default**: Reimpostare i referenzianti al loro valore di default
  - **No Action**: Dare immediatamente errore



# Politiche di risposta del DBMS su violazione vincoli di integrità referenziale

- Es:

```
CREATE TABLE Impiegato (  
    Matricola CHAR(6),  
    Nome CHAR(20) NOT NULL,  
    Cognome CHAR(6) NOT NULL,  
    Dipart CHAR(15),  
    Stipendio NUMERIC(9) DEFAULT 0,  
    PRIMARY KEY (Matricola),  
    FOREIGN KEY (Dipart)  
        REFERENCES Dipartimento(NomeDip)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE,  
    UNIQUE (Cognome, Nome)  
)
```

# Alterazione di schemi in tabelle già esistenti

- **ALTER TABLE NomeTabella**
  - **ADD COLUMN definizioneattributo**
  - **DROP COLUMN nomeattributo**
  - **ALTER COLUMN nomeattributo ...**
  - **ADD CONSTRAINT vincolo**
  - **DROP CONSTRAINT nomevincolo**
- e molto altro... (es: ridenominazione colonne, spostamento ordine colonne, ...): sintassi variabile da DBMS a DBMS ...

# Eliminazione Tabella

## **DROP TABLE nometabella**

- Rimuove una tabella dal sistema (schema e dati): generalmente viene chiesta conferma in modalità interattiva.



# Inserimento dati

```
INSERT INTO nometabella [(col1, col2, col3,...)] VALUES  
    (v1,v2,v3,...), (...)
```

Inserisce una nuova riga nella tabella (se non vengono violati vincoli) assegnando i valori indicati (v1, v2, ...) alle colonne specificate (col1, col2, ...).

**Es:**

```
INSERT INTO Impiegato (Matricola, Nome, Cognome) VALUES  
    ('089342', 'Mario', 'Rossi')
```

- Le colonne non devono necessariamente essere specificate nell'ordine di quelle usate nella definizione della tabella
- I nomi delle colonne possono essere omesse, in questo caso sono usate le colonne usate nella definizione della tabella

# Select: elementi di base

```
SELECT ListaAttributi  
FROM ListaTabelle  
[WHERE Condizione]
```

- Esegue il “prodotto” tra le tabelle indicate in *ListaTabelle* (se più di una)
- Rimuove le righe che non soddisfano la *Condizione*
- Rimuove le colonne che non sono elencate nella *ListaAttributi*

# Lista attributi

- Nel caso la lista attributi sia \* si considerano tutti gli attributi della tabella

- Es:

```
SELECT * FROM Impiegati
```

- Ogni attributo puo' essere anche un'espressione piu' elaborata.

- Es:

```
SELECT Stipendio / 12  
FROM Impiegati  
WHERE Cognome = 'Bianchi'
```

- Per ogni riga, invece di riportare il valore puro di *Stipendio*, ne valuta la divisione per 12 (mesi) per avere lo stipendio mensile.



# Rinominazione Colonna

- Ogni attributo della lista puo' essere rinominato nella tabella risultato restituita dalla SELECT.
- Es:  

```
SELECT Stipendio / 12 AS StipendioMensile  
FROM Impiegati  
WHERE Cognome = 'Bianchi'
```
- Produrrà qualcosa del genere:

StipendioMensile
1200

# Attributi

- Usando l'operatore . (punto) è possibile indicare la tabella di provenienza del campo (particolarmente utile in caso di ambiguità di nomi nel caso di prodotto tra tabelle)

- Es:

```
SELECT Impiegato.Stipendio  
FROM Impiegato  
WHERE Impiegato.Cognome = 'Bianchi'
```

# Lista Tabele

- Nel caso si specifichino più tabelle
  - **SELECT \* FROM Studenti, Esami**
- viene effettuato il prodotto tra le tabelle, ogni riga di una tabella accoppiata con tutte le righe delle altre tabelle

Matricola	Nome
12345	Mario Rossi
25634	Giuseppe Verdi

Matricola	Corso	Voto
12345	Informatica	27
12345	Analisi	28
25634	Fisica	29

Matricola	Nome	Matricola	Corso	Voto
12345	Mario Rossi	12345	Informatica	27
12345	Mario Rossi	12345	Analisi	28
12345	Mario Rossi	25634	Fisica	29
25634	Giuseppe Verdi	12345	Informatica	27
25634	Giuseppe Verdi	12345	Analisi	28
25634	Giuseppe Verdi	25634	Fisica	29



# Liste Tabelle

```
SELECT Nome, Corso, Voto FROM Studenti, Esami  
WHERE Studenti.Matricola=Esami.Matricola
```

Matricola	Nome	Matricola	Corso	Voto
12345	Mario Rossi	12345	Informatica	27
12345	Mario Rossi	12345	Analisi	28
12345	Mario Rossi	25634	Fisica	29
25634	Giuseppe Verdi	12345	Informatica	27
25634	Giuseppe Verdi	12345	Analisi	28
25634	Giuseppe Verdi	25634	Fisica	29

Righe non  
soddisfano  
condizione

Nome	Corso	Voto
Mario Rossi	Informatica	27
Mario Rossi	Analisi	28
Giuseppe Verdi	Fisica	29

Risultato

# Lista Tabelle

- E' possibile attribuire dei “soprannomi” (alias) a ciascuna tabella elencata in lista

- Es:

```
SELECT I.Nome, Cognome, Citta  
FROM Impiegato AS I, Dipartimento AS D  
WHERE Dipart = D.Nome
```

- Si noti che usare il nome tabella o il soprannome è obbligatorio solo nel caso di ambiguità di provenienza di attributi (es: Impiegato.Nome oppure Dipartimento.Nome?)

# Condizione

- Ogni condizione WHERE può essere il risultato di una composizione di più condizioni unite da operatori booleani AND, OR e NOT:

```
SELECT Nome, Cognome
```

```
FROM Impiegato
```

```
WHERE Cognome='Rossi' AND
```

```
(Dipart = 'Amministrazione' OR Dipart ='Produzione')
```

- Trova tutti gli impiegati il cui cognome è Rossi e che lavorano in Amministrazione o in Produzione.



# Condizioni

- Tipicamente per le condizioni semplici si usano gli operatori tradizionali ( $=$ ,  $<$ ,  $>$ ,  $<>$ ,  $<=$ ,  $>=$ ) ma per le stringhe esiste *anche* l'operatore speciale LIKE per la verifica la somiglianza, piuttosto che l'esattezza.
- LIKE supporta due caratteri speciali:
  - “%” significa “qualsiasi ripetizione di carattere”
  - “\_” significa “qualsiasi carattere”

# Condizioni LIKE

Es 1:

```
SELECT *  
FROM Impiegato  
WHERE Cognome LIKE 'Ro%'
```

- Troverebbe tutti gli impiegati il cui cognome comincia per “Ro” (es: Rossi, Romagnoli, ...)

Es 2:

```
SELECT *  
FROM Impiegato  
WHERE Cognome LIKE 'R_ssi'
```

- Troverebbe tutti i cognomi di 5 lettere in cui solo la seconda lettera e' indefinita, mentre le altri sono R, s, s, i (es: Rossi, Russi, Rassi, RAssi, ...)

# Condizioni

- Se in una condizione compare un valore costante:
  - Per le stringhe si usano gli apici singoli (es: ... **WHERE** **Cognome** = 'Rossi')
  - Per i numeri invece si usa la loro rappresentazione senza simboli aggiuntivi (es: ...**WHERE** **Anno** = 1975 **OR** **Temperatura**=22.5 )



# Condizioni

- La presenza o assenza di un valore nullo in un attributo di una riga può essere verificata con l'operatore IS NULL o IS NOT NULL. Es:

```
SELECT Nome, Cognome  
FROM Impiegato  
WHERE StatoCivile IS NOT NULL
```

- Trova tutti i dipendenti che hanno uno stato civile noto (se si immagina di aver rappresentato con NULL l'assenza di tale informazione)

# Condizioni

- Se un valore può essere NULL e non viene verificato con l'operatore IS/IS NOT NULL, la sua presenza insieme ad altre condizioni può causare qualche problema.
- In generale SQL segue la regola dell'inclusione di righe le cui condizioni siano soddisfatte indipendentemente dalla verità o falsità della condizione resa indefinita dal valore nullo.

# Condizioni

- Es:

```
SELECT *  
FROM Impegato  
WHERE Stipendio/12 >= 1000 OR  
StatoCivile = 'CELIBE'
```

- Se una certa riga ha un valore di StatoCivile posto a NULL ma Stipendio/12 maggiore di 1000 euro/mese, SQL includerà la riga dato che, indipendentemente dal valore che potrebbe assumere StatoCivile, la condizione risulterebbe sempre vera (per presenza dell'operatore booleano "OR")



# Gestione duplicati: DISTINCT

- Per ragioni di efficienza, *SQL può mantenere righe duplicate nelle tabelle* (vincoli di integrità permettendo!)
- Per ottenere un comportamento analogo al concetto di relazione matematica (che non ammette duplicati) si può usare l'operatore DISTINCT. Ad esempio:

```
SELECT DISTINCT CittàDiResidenza  
FROM Persona
```

- riporta *una sola volta* ciascuna città oggetto di residenza delle persone presenti nella tabella Persona. La stessa query senza la parola DISTINCT avrebbe riportato ciascuna città *tante volte* quante è oggetto di residenza di una persona.

# UPDATE

- Istruzione UPDATE serve ad aggiornare le righe di una tabella/tabelle:
  - `UPDATE ListaTabelle SET Attributo1=espressione, Attributo2=espressione,... [WHERE condizione]`
- La condizione è analoga a quella usata per SELECT, se omessa viene applicata a tutta la tabella
- Es:
  - `UPDATE Persone SET eta=eta+1;`
    - Incrementa di uno l'età di tutte le persone nella tabella
  - `UPDATE Impiegati SET Stipendio=Stipendio+150 WHERE annoAssunzione=2006;`
    - Incrementa lo stipendio di 150 euro per gli assunti nel 2006

# DELETE

- Istruzione DELETE cancella righe dalla tabella:
  - **DELETE FROM *ListaTabelle* [WHERE *condizione*]**
- Es:
  - **DELETE Persone WHERE dataMorte IS NOT NULL;**

# Join

- L'operazione di join unisce i dati contenuti in piu' tabelle

- Es:

```
SELECT * FROM Studenti, Esami  
WHERE Studenti.Matricola =  
Esami.Matricola
```

- Questo tipo di operazione si chiama **JOIN** (unione) **INTERNO** e la condizione sottolineata si chiama *condizione di join*.



# Join

- SQL fornisce un'espressione apposita per indicare esplicitamente la condizione di join senza inserirla nella clausola WHERE:

```
SELECT ...
```

```
FROM Tabella1 INNER JOIN Tabella2 On
```

```
Tabella1.Attributo1 = Tabella2.Attributo2
```

```
WHERE altre condizioni...
```

- Questa espressione non è solo più chiara per chi legge l'istruzione di SELECT, ma informa anticipatamente il sistema che deve effettuare un join: in questo modo il database può ottimizzare internamente l'interrogazione, risultando più veloce.
- Nota: anche il join interno è ovviamente applicabile su più di due tabelle!

# Join

- Es:

```
SELECT I.Nome, Cognome, Città, D.Indirizzo  
FROM Impiegato As I  
INNER JOIN  
DIPARTIMENTO As D  
ON I.Dipart = D.Nome
```

- lega impiegati e dipartimenti tramite i campi Dipart e Nome

# Join esterno

- Con il join interno può capitare che alcune righe di una delle due tabelle non finiscano nella tabella risultato a causa di una assenza di legame con righe dell'altra tabella.
- Con il **JOIN ESTERNO** è possibile chiedere a SQL di inserire comunque le righe scartate, associandole a valori NULL in corrispondenza degli attributi dell'altra tabella.

# Join esterno

- Esistono tre tipi di join esterno:
  - Sinistro (left outer join)
  - Destro (right outer join)
  - Completo: sinistro e destro (full outer join)
- Nel primo caso sono i valori della tabella di sinistra che vengono riportati al completo nella tabella risultato; nel secondo sono quelli della tabella di destra; nell'ultimo caso vengono sempre riportati TUTTE le righe



# Join esterno

- Sintassi:

```
SELECT ...  
FROM Tabella1 LEFT|RIGHT|FULL OUTER JOIN Tabella2 ON  
Tabella1.Campo1 = Tabella2.Campo2  
WHERE ...
```

- Es:

```
SELECT *  
FROM Persone Right Outer Join Citta On  
Persone.Citta = Citta.Nome
```

- fornisce *tutte* le città della tabella Città, anche quelle in cui non esiste nessuna persona residente.
- Nota 1: la parola *OUTER* è facoltativa.
- Nota 2: si osservi come gli outer join non siano esprimibili con SELECT tradizionali (a meno di altri costrutti che ancora non sono stati esposti, e comunque sempre in modo molto innaturale e macchinoso).

# Criteri Ordinamento

- E' possibile usare la parola chiave ORDER BY per richiedere l'ordinamento delle righe-risultato secondo il valore di uno o più attributi
- Es:  

```
SELECT *  
FROM Persone  
ORDER BY Città, Nome
```
- Ordina le righe in ordine crescente dell'attributo città e, a parità di città, in ordine crescente di Nome della persona
- Nota: per ciascun attributo è possibile specificare ASC o DESC per indicare il tipo di ordinamento desiderato (crescente o decrescente)

# Operatori di aggregazione

- SQL mette a disposizione un set di operatori per tutte quei casi in cui si ha la necessità di valutare una proprietà che dipende da un *insieme* di righe, piuttosto che da una singola riga. Questi operatori sono:
  - COUNT
  - SUM
  - MIN e MAX
  - AVG

# Operatore COUNT

- Conta e restituisce il numero di righe ottenute dal risultato di una SELECT
- Es:  

```
SELECT Count(*)  
FROM Persone  
WHERE Città='FIRENZE'
```
- Conta il numero di righe che hanno 'FIRENZE' come città di residenza => restituisce il numero di residenti a Firenze.



# Operatore COUNT

- Usato con DISTINCT conta i valori distinti (quindi senza considerare le ripetizioni)
- Es:  

```
SELECT COUNT (DISTINCT Città)  
FROM Persone
```
- Restituisce il numero di Città (contandole ciascuna una sola volta) per le quali esiste almeno un residente.

# Operatore SUM

- SUM valuta la somma dei valori presenti su tutte le righe in corrispondenza di un attributo
- Es:  

```
SELECT SUM(Importo)  
FROM Spese  
WHERE CodiceAppartamento = 5
```
- Restituisce la somma di tutti gli importi presenti nelle righe della tabella Spese attribuite all'appartamento di codice = 5.

# Operatori MIN/MAX

- Restituiscono, rispettivamente, il valore minimo o massimo presente per un attributo valutato tra tutte le righe della tabella risultato della SELECT
- Es:

```
SELECT MAX(Importo)
```

```
FROM Spese
```

```
WHERE CodiceAppartamento = 5
```

- restituisce la spesa di importo massimo dell'appartamento di codice 5

# Operatore AVG

- AVG calcola il valore medio (aritmetico) per un attributo, valutandolo tra tutti i valori di quell'attributo presenti nelle righe della tabella risultato della SELECT

- Es:

```
SELECT AVG(Importo)
```

```
FROM Spese
```

```
WHERE CodiceAppartamento = 5
```

- Calcola l'importo medio delle spese per l'appartamento 5.



# GROUP BY

- Spesso risulta utile applicare gli operatori di aggregazione (SUM, MIN, MAX, ...) a sottogruppi delle righe della tabella, piu' che all'intera tabella.
- Per questo scopo la sintassi della SELECT è stata estesa con la clausola **GROUP BY**, che crea dei gruppi di righe all'interno della tabella risultato del prodotto.

# GROUP BY

- Sintassi semplificata:

**SELECT *ListaAttributi***

**FROM *ListaTabelle***

**WHERE *Condizione/i***

**GROUP BY *ListaAttributiDiRaggruppamento***

- Il sistema esegue il prodotto tra le tabelle indicate nella clausola FROM, scarta le condizioni che non soddisfano la condizione WHERE e poi crea delle sottotabelle immaginarie raggruppando valori uguali degli attributi indicati a fianco della GROUP BY

# GROUP BY

- Es.:

```
SELECT Dipart, SUM(Stipendio)
FROM Impiegato
GROUP BY Dipart
```

- Di seguito è mostrata la tabella Impiegato:

**Impiegato**

Nome	Cognome	Dipart	Ufficio	Stipendio
Mario	Rossi	Amministrazione	10	22
Carlo	Bianchi	Produzione	20	18
Giuseppe	Verdi	Amministrazione	20	20
Franco	Neri	Distribuzione	16	22
Carlo	Rossi	Direzione	14	40
Lorenzo	Lanzi	Direzione	7	36
Paola	Biscelgia	Amministrazione	75	20
Marco	Franco	Produzione	20	23



# GROUP BY

- La SELECT:
  - Esegue il prodotto (ma qui esiste solo una tabella!)
  - Applica la condizione WHERE alle righe (ma qui la condizione manca!)
  - Raggruppa *idealmente* le righe a parità di valore dell'attributo Dipart:

Nome	Cognome	Dipart	Ufficio	Stipendio
Mario	Rossi	Amministrazione	10	22
Giuseppe	Verdi	Amministrazione	20	20
Paola	Biscelgia	Amministrazione	75	20
Carlo	Rossi	Direzione	14	40
Lorenzo	Lanzi	Direzione	7	36
Franco	Neri	Distribuzione	16	22
Carlo	Bianchi	Produzione	20	18
Marco	Franco	Produzione	20	23



# GROUP BY

- (continua)
  - Elimina le colonne non richieste
  - Per ciascun gruppo riporta il valore comune del campo Dipart e la somma dei valori di Stipendio all'interno del gruppo:

Dipart	SUM(Stipendio)
Amministrazione	62
Direzione	76
Distribuzione	22
Produzione	41

- Nota: se viene usato GROUP BY, SQL impone di utilizzare nella ListaAttributi solo gli attributi di raggruppamento oppure operatori aggregati sugli altri attributi.

# GROUP BY

- Sintassi completa:

**SELECT *ListaAttributi***

**FROM *ListaTabelle***

**WHERE *Condizione/i sulle righe***

**GROUP BY *ListaAttributiDiRaggruppamento***

**HAVING *Condizione/i sui gruppi***

- La clausola aggiuntiva HAVING permette di esprimere condizioni aggiuntive per scartare gruppi che non interessano, analogamente a quanto succede per le righe con la WHERE

# GROUP BY

- Es:

```
SELECT Dipart, SUM(Stipendio) As SommaStipendi  
FROM Impiegato  
GROUP BY Dipart  
HAVING SUM(Stipendio) > 50
```

- E' analoga alla precedente, salvo il fatto che verranno scartati i gruppi di righe la cui somma degli stipendi e' inferiore o uguale a 50.
- Nota: E' ovviamente possibile includere un ordinamento finale su un attributo di gruppo attraverso una ORDER BY in modo del tutto analogo a quanto visto in precedenza (Es: ORDER BY Dipart)

# Query Insiemistiche

- Sono supportate tre operazioni su insiemi:

- UNION (unione)
- INTERSECT (intersezione)
- EXCEPT (differenza)

- Sintassi:

***SelectSQL1 UNION | INTERSECT | EXCEPT [ALL]  
SelectSQL2***



# Query Insiemistiche

- Es:

```
SELECT Nome  
FROM Impiegato
```

**UNION**

```
SELECT Cognome  
FROM Impiegato
```

- Crea un'unica tabella da una colonna con l'insieme dei nomi e dei cognomi, riportati una solo volta
- Nota: Usando l'operatore ALL si permette la ripetizione degli elementi.

# Query Insiemistiche

- Se nel precedente esempio si cambia UNION con INTERSECT, ottengo solo i nomi che sono presenti anche come cognome.
- Ancora, se UNION viene sostituito con EXCEPT, vengono considerati *solo* i nomi che NON compaiono come cognomi.

# Query Nidificate

- Ultima complicazione per estendere la potenza della SELECT: accettare nella condizione WHERE anche confronti tra un singolo valore e il risultato di un'ulteriore SELECT (che viene detta *nidificata* o *subordinata*).
- Es:  

```
SELECT * FROM Impiegato  
WHERE Dipart IN ( SELECT Nome FROM  
Dipartimento WHERE Città='Firenze' )
```
- Recupera tutti gli impiegati che lavorano in dipartimenti situati a Firenze.
- Nota: La stessa query poteva essere espressa mediante una join.

# Query Nidificate

- Esistono due principali tipi di interrogazioni subordinate:
  - “**insiemistiche**”: che ritornano un insieme di valori (=una tabella composta da *una sola colonna*)
  - “**scalari**”: che ritornano un unico valore (=una tabella composta da una sola colonna e *una sola riga*)



# Sottointerrogazioni Scalari

- Dal momento che restituiscono un solo valore, questo può essere confrontato con una qualsiasi espressione dell'interrogazione principale

- Es:

```
SELECT Nome,Dipart FROM Impiegato  
WHERE Stipendio = (SELECT MAX(Stipendio) FROM  
Impiegato )
```

# Sottointerrogazioni Insiemistiche

- Operatore insiemistico **IN**: verifica se un certo elemento appartiene ad un insieme di valori (=> anche ad un insieme prodotto da una SELECT subordinata)

- Es: (già presentato)

```
SELECT * FROM Impiegato
WHERE Dipart IN ( SELECT Nome
                  FROM Dipartimento
                  WHERE Città='Firenze' )
```

- Nota: la SELECT subordinata deve restituire una tabella composta da una sola colonna!

# Sottointerrogazioni Insiemistiche

- Operatori insiemistici **ANY** e **ALL**: permettono un confronto tra un valore e un elenco di valori presente in un insieme ( $\Rightarrow$  anche l'insieme risultato di una SELECT subordinata). Forma dell'espressione:

***valore*** = | < | > | <= | >= | <> **ANY** | **ALL** *insieme*

- In particolare:
  - ANY verifica che la condizione sia soddisfatta in *almeno un valore* dell'insieme
  - ALL verifica che la condizione sia soddisfatta da *tutti i valori* dell'insieme

# Sottointerrogazioni Insiemistiche

- Es:

```
SELECT Dipart FROM Impiegato
WHERE Stipendio >= ALL ( SELECT
    Stipendio FROM Impiegato )
```

- seleziona l'impiegato che ha lo stipendio  $\geq$  di tutti gli stipendi degli impiegati dalla tabella.
- Confrontare con la precedente soluzione proposta che usava una subquery scalare:

```
SELECT Dipart FROM Impiegato
WHERE Stipendio = (SELECT
    MAX(Stipendio) FROM Impiegato )
```



# Sottointerrogazioni Insiemistiche

- Operatore insiemistico **EXISTS**: non ha bisogno di espressioni di confronto e restituisce vero se esiste almeno un elemento all'interno dell'insieme fornitogli come argomento
- Es:

```
SELECT * FROM Persona AS P
WHERE NOT EXISTS (SELECT *
FROM Persona AS P1
WHERE P1.Nome = P.Nome AND
P1.Cognome = P.Cognome AND
P1.CodFisc <> P.CodFisc )
```

- Restituisce tutte le persone che NON hanno degli omonimi.

# Vincoli Integrita'

- SQL mette a disposizione la parola chiave CHECK per esprimere un vincolo di integrità che non sia compreso tra quelli elementari già esposti (Not Null, Primary Key, Refereces, Foreign Key)

```
CREATE TABLE NomeTabella (  
...  
Attributo Dominio CHECK condizione  
...  
)
```

- “*Condizione*” può essere una qualsiasi condizione della stessa forma strutturale di una qualunque altra espressione utilizzabile in una WHERE, *compreso l'utilizzo di SELECT nidificate*.

# Vincoli Integrita'

- Es:

```
CREATE TABLE Impiegato (  
  Matricola CHAR(6) CHECK ( Matricola IS NOT NULL AND  
    2 > (SELECT COUNT(*)  
      FROM Impiegato AS I  
      WHERE Matricola = I.Matricola)),  
  Cognome CHAR(20) CHECK (Cognome IS NOT NULL),  
  Nome CHAR(20) CHECK (Nome IS NOT NULL),  
  Dipart CHAR(15) CHECK (Dipart IN (SELECT NomeDip  
    FROM Dipartimento))  
)
```

- crea la tabella Impiegato esprimendo ogni vincolo con una apposita CHECK anziché utilizzate le apposite parole chiave dei vincoli base.

- Es:

**Voto INTEGER CHECK Voto >=18 AND Voto <=30**

# Viste

- **Vista:** tabella virtuale il cui contenuto dipende dalle altre tabelle di una base di dati.
- Finalità delle viste:
  - Rendere le interrogazioni più semplici, ipotizzando l'esistenza di tabelle a struttura semplificata che in realtà non esistono.
  - Mostrare a utenti diversi un insieme di dati ridotto rispetto a quelli realmente presenti sulla base di dati
- Creazione:

**`CREATE VIEW NomeVista AS SelectSQL`**



# Viste

- Es:

```
CREATE VIEW ImpiegatiAmministrativi(Matricola, Nome,  
Cognome, Stipendio) AS  
SELECT Matricola, Nome, Cognome, Stipendio  
FROM Impiegato  
WHERE Dipart = 'Amministrazione' AND Stipendio > 5
```

- Da adesso in poi la vista può essere usata a tutti gli effetti come se fosse una tabella:

```
SELECT * FROM ImpiegatiAmministrativi  
WHERE Cognome = 'Rossi'
```

- Oppure:

```
UPDATE ImpiegatiAmministrativi SET Stipendio = Stipendio  
+ 1
```

# Viste

- **Nota 1:** solo le viste costruite a partire *da una sola* tabella sottostante possono essere utilizzate in un contesto di modifica (ex: UPDATE). Quando le tabelle coinvolte sono più di una, la vista può essere utilizzata solo in lettura (=> solo via SELECT)
- **Nota 2:** una vista viene sempre costruita dinamicamente (=“al volo”) a partire dalle tabelle sottostanti: nessuna copia dei dati viene mantenuta dal DBMS => è possibile eliminare (DROP) una vista senza perdere alcun dato delle tabelle sottostanti.

# Indici

- **Indice:** struttura interna al DBMS che permette di recuperare più rapidamente le informazioni ricercate su un insieme di attributi attraverso il mantenimento di un ordinamento dei valori di tali attributi
- Creazione di un indice:

```
CREATE [UNIQUE] INDEX NomeIndice ON  
  NomeTabella(ListaAttributi)
```

# Indici

- Es: data la tabella PERSONA(CodFisc, Nome, Cognome, DataNascita)
- Viene creato un indice su CodiceFiscale nel seguente modo:  
`CREATE UNIQUE INDEX IndiceCodFiscale ON PERSONA(CodFisc)`
- In tal modo, una query come quella seguente potrà essere eseguita molto più rapidamente, dato che il sistema memorizza la posizione di ogni codice fiscale che compare nella tabella (=> non sono necessarie scansioni sequenziali della stessa)

```
SELECT * FROM PERSONA
```

```
WHERE CodFisc = 'CFGARN65T12D612Y'
```



# Indici

- **Nota 1:** gli indici velocizzano drasticamente le interrogazioni, ma penalizzano le operazioni di inserimento, modifica e cancellazione, dato che l'indice deve essere mantenuto aggiornato dal DBMS
- **Nota 2:** conviene associare indici solo a campi che vengono utilizzati più spesso come elementi di ricerca nelle query SQL
- **Nota 3:** SQL indicizza automaticamente i campi dichiarati come chiavi primarie, per l'importanza che essi rivestono nei join tra tabelle

# Utenti e Controllo Accessi

- Ogni DBMS relazionale mette a disposizione meccanismi di protezione basati sull'identificazione dell'utente:
  - Identificazione tramite funzionalità del sistema operativo
  - Identificazione interna al DBMS, con propri utenti
- Nota: l'identificazione avviene al momento della fase di connessione al DBMS

# Utenti e Controllo Accessi

- Controllo di accesso significa:
  - Garantire o proibire una **tipologia di operazioni** (“**privilegi**”)
  - ad uno specifico **utente**
  - in una o più **risorse** del DBMS (es: tabelle)

*GRANT **privilegi** ON **risorsa** TO **utenti***
- Abilita un insieme di privilegi sulla risorsa indicata all'insieme di utenti desiderato.

# Utenti e Controllo Accessi

- Principali tipi di privilegi:
  - *Insert*: possibilità di inserimento di nuovi elementi nella risorsa (applicabile a: *tabelle* e *attributi*)
  - *Update*: possibilità di modifica di valori della risorsa (applicabile a: *tabelle* e *attributi*)
  - *Delete*: possibilità di cancellazione di elementi (applicabile a: solo *tabelle*)
  - *Select*: possibilità di lettura di valori (applicabile a: solo *tabelle*)

• Es:

```
GRANT Select ON Dipartimento TO Stefano
```

• Es:

```
GRANT All Privileges ON Impiegati TO Paolo, Riccardo
```



# Utenti e Controllo Accessi

- Revoca di privilegi:
  - **REVOKE privilegi ON risorsa FROM utenti**
- Nota su GRANT: aggiungendo in coda al comando la voce **WITH GRANT OPTION**,
- gli utenti hanno la possibilità di garantire gli stessi privilegi ad altri.
- Ulteriore nota: il creatore(“proprietario”) di una tabella ha sempre tutti i diritti garantiti su tale risorsa.

# PHP & MySQL

- PHP fornisce una serie di funzioni per accesso a database
- In particolare sono presenti funzioni per uso di MySQL ma anche di postgresSQL, Microsoft SQL, DB2 IBM, Oracle, SQLite, ...

# mysql\_connect

- La funzione **mysql\_connect** si usa per connettersi a un database mysql fornendo nome della macchina e le credenziali (utente+password) per l'accesso:  

```
resource mysql_connect(string nomehost,  
                        string username, string password)
```
- Se i parametri sono omessi, valori di default sono presi da "php.ini"
- Ritorna una risorsa per usare la connessione, in caso di errore ritorna FALSE e si può usare la funzione **mysql\_error()** per ottenere il messaggio di errore.

- Es:

```
$r=mysql_connect("localhost", "utente",  
                "abracadabra") or die('fallita connessione a db:' .  
                mysql_error());
```

# mysql\_select\_db

- La funzione ***mysql\_select\_db*** si usa per impostare il database da usare per effettuare le query SQL:  
`bool mysql_select_db(string db_name [,resource link ])`
  - Il primo parametro indica il nome del database da usare.
  - Il secondo parametro indica il link da usare e se viene omesso usa per default l'ultimo link aperto tramite ***mysql\_connect(...)***

- ***Es:***

```
$r=mysql_connect("localhost", "root", "nessuna");  
mysql_select_db("metodi"); //usa db metodi  
...
```



# mysql\_query

- La funzione ***mysql\_query*** serve a inviare al server SQL una query SQL:

```
resource mysql_query(string query [, resource link])
```

- Il primo parametro indica la query da eseguire
- Il secondo parametro indica l'eventuale link da usare per la query, se omissso usa l'ultimo link aperto con *mysql\_connect*
- Il valore di ritorno viene utilizzato per ottenere i dati risultato della query, ritorna false in caso di fallimento della query.

- Esempio:

```
$link = mysql_connect("localhost", "user", "pwd") or  
die(mysql_error());
```

```
mysql_select_db("metodi");
```

```
$r = mysql_query("SELECT * FROM Studenti") or  
die(mysql_error());
```

...

# mysql\_fetch\_array

- La funzione ***mysql\_fetch\_array*** serve a prelevare i dati di una riga del risultato di una query effettuata con ***mysql\_query***, inoltre fa in modo che la chiamata successiva a funzione di fetch prelevi i dati della riga successiva (se presente):

```
array mysql_fetch_array(resource result, int type =  
    MYSQL_BOTH)
```

- Il primo parametro è il risultato ottenuto da una `mysql_query`
- Il secondo parametro indica come costruire l'array contenente i dati della riga:
  - `MYSQL_ASSOC` indica di creare un array associativo che associa il nome del campo al valore
  - `MYSQL_NUM` indica di creare un array che associa alla posizione del campo il suo valore
  - `MYSQL_BOTH` indica di creare un array che combina `MYSQL_ASSOC` e `MYSQL_NUM`, questo è il default

# Mysql\_fetch\_array

- Esempio:

```
mysql_connect("localhost", "user", "pwd");  
mysql_select_db("metodi");  
$r=mysql_query("SELECT Nome,Cognome FROM Studenti");  
while($a=mysql_fetch_array($r))  
{  
    echo $a["Cognome"].", ". $a["Nome"]."<br>";  
    //echo "$a[1], $a[0]<br>";  
}
```

# mysql\_escape\_string

- Quando si fanno delle query è possibile che siano presenti caratteri come ' nelle stringhe SQL, in questo caso la ' va preceduta da \ per annullare il suo significato di chiusura stringa
- Esempio:

...

```
$t="c'e";
```

```
$query="SELECT * from Libri where titolo='$t';
```

```
mysql_query($query) or die(mysql_error());
```

...

**Produce la query:**

```
SELECT * from Libri where titolo='c'e'
```

Che non è sintatticamente corretta, invece

```
SELECT * from Libri where titolo='c\'e\'
```

È sintatticamente corretta



# mysql\_escape\_string

- Per ovviare al problema precedente si può usare la funzione ***mysql\_escape\_string()*** che data una stringa fa l'escaping dei caratteri che potrebbero dare problemi in una query
- Esempio:

```
...  
$t="c'e";  
$query="SELECT * from Libri where titolo=" .  
    mysql_escape_string($t) . "'";  
mysql_query($query) or die(mysql_error());  
...
```

# Altre funzioni

- **int mysql\_num\_rows(\$result)**
  - Ritorna il numero di righe del risultato di una query
- **int mysql\_affected\_rows([\$link])**
  - Ritorna il numero di righe su cui l'ultima query ha avuto effetto, da usare con query INSERT, UPDATE, DELETE, REPLACE
- **int mysql\_num\_fields(\$result)**
  - Ritorna il numero di colonne del risultato di una query
- **string mysql\_field\_name(\$result, \$col)**
  - Ritorna il nome del campo del risultato di una query per la colonna indicata da \$col (parte da zero)
- **string mysql\_field\_len(\$result, \$col)**
  - Ritorna la lunghezza del campo del risultato di una query per la colonna indicata da \$col.
- **string mysql\_field\_type(\$result, \$col)**
  - Ritorna il tipo del campo del risultato di una query per la colonna indicata da \$col.
- **mysql\_close([\$link])**
  - Chiude la connessione con il server SQL, se non indicato chiude l'ultima connessione aperta con mysql\_connect(). Le connessioni vengono comunque chiuse al termine dello script

# Esempio

```
mysql_connect("localhost", "user", "pwd");
mysql_select_db("metodi");
$r=mysql_query("SELECT * FROM Studenti");
$l=mysql_num_fields($r);
echo "<table><tr>";
for($i=0; $i<$l; $i++)
{
    echo "<td><b>".mysql_field_name($r, $i)."</b><br>";
    echo mysql_field_type($r, $i). "(";
    echo mysql_field_len($r, $i).")</td>";
}
echo "</tr>";
while($a=mysql_fetch_array($r))
{
    echo "<tr>"
    for($i=0; $i<$l; $i++)
        echo "<td>".$a[$i]."</td>";
    echo "</tr>";
}
echo "</table>";
```

## Link utili

- <http://www.w3.org/XML>
- <http://www.w3.org/2004/11/uri-iri>  
[pressrelease.html.en](http://www.w3.org/2004/11/uri-iri/pressrelease.html.en)
- [http://www.w3.org/TR/wsdl#\\_document-s](http://www.w3.org/TR/wsdl#_document-s)
- <http://www.w3.org/2004/02/skos/>
- <http://www.w3.org/TR/wsdl>



# Programmazione in PHP e CMS

Sistemi distribuiti, Corso di Laurea in Ingegneria

*Ing. Michela Paolucci*

Department of Information Engineering

University of Florence

Via S. Marta 3, 50139, Firenze, Italy

tel: +39-055-2758515, fax: +39-055-2758570

**Lab: DISIT, Sistemi Distribuiti e Tecnologie Internet**

**<http://www.disit.dinfo.unifi.it/>**

[michela.paolucci@unifi.it](mailto:michela.paolucci@unifi.it)

<http://www.disit.org/nesi>, <http://www.dsi.unifi.it/~paolucci>,